Volume 1 2021 Number 1

JOURNAL OF BEHAVIORAL DATA SCIENCE

ISBN: 2575-8306 (Print) 2574-1284 (Online)

https://isdsa.org/jbds

JOURNAL OF BEHAVIORAL DATA SCIENCE

No Publication Charge and Open Access

jbds@isdsa.org

List of Articles

Journal of Behavioral Data Science, 2021, *1* (1), 1–16.
DOI: https://doi.org/10.35566/jbds/v1n1/p1

What is Data Science? An Operational Definition based on Text Mining of Data Science Curricula

Zhiyong Zhang[1] and Danyang Zhang[2]

[1] University of Notre Dame
zzhang4@nd.edu
[2] University of Texas–Austin
danyang.zhang@utexas.edu

Abstract. Data science has maintained its popularity for about 20 years. This study adopts a bottom-up approach to understand what data science is by analyzing the descriptions of courses offered by the data science programs in the United States. Through topic modeling, 14 topics are identified from the current curricula of 56 data science programs. These topics reiterate that data science is at the intersection of statistics, computer science, and substantive fields.

Keywords: Data Science · Topic Modeling · Data Science Curriculum.

1 Introduction

Data science has been a buzzword in the past two decades. However, the exact meaning of data science has never been clear. In a statement by American Statistical Association (ASA), it states "there is not yet a consensus on what precisely constitutes data science" (Van Dyk et al., 2015). Hayashi (1998) is probably the first formal attempt to define data science although the history of data science practice is considerably longer (Donoho, 2017; Tukey, 1962).[1] He argued that "the aim of data science is to reveal the features or the hidden structure of complicated natural, human and social phenomena with data" and data science consists of "design for data, collection of data, and analysis on data." To many, this sounds like the characteristics of applied statistics (e.g., Broman, 2013; Silver, 2013). Not surprisingly, some have also argued that data science is actually different from statistics. For example, Dhar (2013) pointed out that data science is different from statistics in terms of data types and skills required. The

[1] Tukey has used the term "data analysis" in his writing that is conceptually similar to what data science does. Naur (1966) coined the term "datalogy" to call "the science of the nature and use of data" and Naur (1974) provided a more detailed treatment of data largely from a computer science perspective.

current view of data science aligns more closely with what Cleveland (2001) has described – data science consists of 25% Multidisciplinary Investigations, 20% Models and Methods for Data, 15% Computing with Data, 15% Pedagogy, 5% Tool Evaluation, and 20% Theory. Regardless of how it is perceived, data science is now widely accepted as its own paradigm (Hey, Tansley, & Tolle, 2009).

Many data science degree programs emerged in the past few years. The Institute for Advanced Analytics (IAA) at North Carolina State University tracks the master's degrees in Data Science at universities based in the United States. By its count, there are 78 data science programs in 2020.[2] From a practical point of view, it is probably more informative to understand what data science offers and what it is constituted than its exact definition that might not be possible at all. In the same ASA statement, Van Dyk et al. (2015) identified three foundations to data science:

> (i) *Database Management enables transformation, conglomeration, and organization of data resources.*
> (ii) *Statistics and Machine Learning convert data into knowledge.*
> (iii) *Distributed and Parallel Systems provide the computational infrastructure to carry out data analysis.*

In a review of the history of data science, Donoho (2017) coined the "Greater Data Science" field with six divisions: data exploration and preparation, data representation and transformation, computing with data, data modeling, data visualization and presentation, and science about data science. Some empirical studies also tried to understand what skills and knowledge are needed in jobs (e.g., Cegielski & Jones-Farmer, 2016) and taught in degree programs (e.g., Gorman & Klimberg, 2014; Song & Zhu, 2016).

More recently, Fayyad and Hamutcu (2020) proposed a "Data Science Knowledge Framework" aiming to support industry standardization and building measurement and assessment methodologies for data science professionals. The framework identified two domains in analytics and data science: Science and Math, and Programming and Technology. Within the Science and Math domain, they identified the following seven fields: Scientific Method, Mathematics, Computer Science, Statistics, Operations Research & Optimization, Data Preparation and Exploration, and Machine Learning. The Programming and Technology domain has four fields: General Purpose Computing, Scientific Computing, Database & Business Intelligence, and Big Data. Fayyad and Hamutcu (2020) also provided a list of subjects for each field with example topics.

However, Fayyad and Hamutcu (2020) did not provide much empirical support to their knowledge framework. The existing empirical studies (e.g., Gorman & Klimberg, 2014; Song & Zhu, 2016) on data science curricula were conducted several years ago without considering the newly emerged programs. The goal of this study is to empirically examine the current data science programs to hopefully better understand and define what data science is. The rest of this paper

[2] We consider the data analytics and business analytics programs as different from data science programs.

is structured as follows. In Section 2, we present our data collection method and data analysis procedure. In Section 3, we report the results from our data analysis. In Section 4, we discuss our findings.

2 Methods

2.1 Data Collection

IAA keeps an up-to-date track of graduate degree programs in analytics, business analytics, and data science offered in the US.[3] From it, we identified 74 programs from 74 universities, one program from each university, with the term "data science" in their names.[4] The actual names of the programs have 17 different varieties such as M.S. in Data Science, M.S.E. in Data Science, M.S. in Computational Data Science, and M.S. in Data Science and Business Analytics. Many data science programs offer different concentrations. For example, Depaul University started its M.S. in Data Science program in 2010 and now has four concentrations: Computational Methods, Health Care, Hospitality, and Marketing.

For each program, we looked through its website and downloaded the information on the courses offered and the description of each course in one of the following two ways. For the majority of the programs, we used Python to download the course information automatically. For the rest, we saved the information manually.

2.2 Data Preprocessing

The 74 programs offered a total of 2,022 courses after removing the same courses listed in different concentrations by the same programs. Different programs can offer the courses with the same names. For example, *Machine learning*, *Data visualization*, and *Data mining* are offered by 28, 19, and 18 programs, respectively, with the exact same names. However, the contents taught in these courses can be different. Only 58 of the 74 programs provided descriptions of the courses they offered at the time of our data collection. In total, 1,383 courses were found to have description information. For some courses, the descriptions were very brief. For example, for one course *Scripting Languages*, its description was "Survey of current business analytics scripting languages." In this study, we removed such courses with short or uninformative descriptions, which eventually resulted in 1,276 courses from 56 programs.

Typical text data preprocessing steps (e.g., Hickman, Thapa, Tay, Cao, & Srinivasan, 2020; Vijayarani, Ilamathi, Nithya, et al., 2015) were taken to prepare the course descriptions for further analysis. First, all words were converted to lower cases and all numbers were removed. Second, we replaced abbreviations

[3] `https://analytics.ncsu.edu/?page_id=4184`

[4] After our data analysis, IAA added four more programs from Old Dominion University, University of Colorado Boulder, University of Miami, and Utah State University.

such as "GIS" with "geographic information systems", and "ML" with "maximum likelihood" so that the same forms of terms were used in all descriptions. Third, we combined some terms with the same or similar meaning such as both "C" and "C++" to "cprogram" and "SQL", "MySQL" and "NoSQL" to "sql". However, we did not conduct word-stemming except for changing all the words in the plural forms to their singular forms because different forms of the words might have different meanings. Fourth, we removed common stopwords such as "a", "the", "about", and "very". Some frequently used words such as "students", "semester", and "assignment" in course descriptions were not conventionally considered as stopwords. However, they did not provide useful information and, therefore, were removed before analysis.

2.3 Data Analysis

With the preprocessed data, we conducted both term frequency analysis and topic modeling.

2.3.1 Term Frequency Analysis. We first tokenized the course descriptions into individual words and analyzed the frequency of each word. A large frequency shows that a word is more frequently used in the course descriptions and indicates the importance of a topic that the word is associated with. After that, we investigated the frequency of short phrases including two-word phrases such as "data mining" and "machine learning", three-word phrases such as "natural language processing" and "support vector machine", and four-word phrases such as "Markov chain Monte Carlo" and "relational database management system".

2.3.2 Topic Modeling. Topic modeling or topic models can be used to investigate the topics and associated words through mining text information. We used topic modeling to identify the common topics in courses offered in data science programs. Latent Dirichlet allocation (LDA; Blei, Ng, & Jordan, 2003) is probably the most widely used method in topic modeling that allows the observed text, in our case the course descriptions, to be explained by latent topics. In LDA, each course description can be assumed as a mixture of a small number of topics, and each word's presence in the description is associated with one of the topics.

One may argue that the name of a course would summarize the main topic of the course. However, it is not necessarily the case based on our quick analysis of the course descriptions. For example, a course was named "Advanced Data Analysis." First, the name itself was not informative. Second, its description included topics on "data visualization techniques", "dimension reduction techniques", and the use of "computer packages." As we will show, these can be viewed as three different topics. Through LDA, we explored how many common topics the courses from many different data science programs cover.

Suppose there are a total of K topics in all courses. For a given course, it can consist of one or all of the K topics with different probabilities. Let z_{km} be the

kth ($k = 1, \ldots, K$) topic in the mth ($m = 1, \ldots, M$) course. z_{km} takes a value between 1 and K. The topic from which a word n associated with is assumed to be generated from categorical distribution

$$z_{mn} \sim Cat(\boldsymbol{\theta}_m)$$

with the topic probability $\boldsymbol{\theta}_m = (\theta_{m1}, \theta_{m2}, \ldots, \theta_{mK})'$ for the course m. Note that $\sum_{k=1}^{K} \theta_{mk} = 1$. For example, if there are two topics, $K = 2$. Let $w_{mn}, n = 1, \ldots, N_m, m = 1, \ldots, M$, be the nth word and N_m be the total number of words in the mth course description. w_{mn} would take a value between 1 and V with V being the total number of unique words used in all the course description. LDA specifies that

$$w_{mn}|z_{mn} = k \sim Cat(\boldsymbol{\beta}_k)$$

where $\boldsymbol{\beta}_k = (\beta_{k1}, \beta_{k2}, \ldots, \beta_{kV})'$ is the probability that a word is used given the topic k is discussed in a course.

The parameters $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ in LDA models are typically not known and need to be estimated. Both frequentist and Bayesian methods are available to estimate the parameters. For example, Blei et al. (2003) proposed both efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes parameter estimation. In this study, we used the Bayesian method based on Gibbs sampling for our data analysis.

In topic modeling, the number of topics is often unknown and needs to be determined. In this study, c-fold cross-validation (CV) was used. The basic idea of CV is to divide the data into c folds, or c subsets. Each time, one uses $c-1$ folds of data to fit a topic model and then uses the left out fold of data to calculate the statistic–perplexity–to evaluate the model fit (Grün & Hornik, 2011). This can be done for different numbers of topics and the model with the close-to-smallest perplexity can be chosen as the one with the optimal number of topics.

Although the LDA was initially conducted on individual words, research has shown that including phrases of a sequence of words can lead to improved topic quality (e.g., Lau, Baldwin, & Newman, 2013; Nokel & Loukachevitch, 2015). Therefore, in our study, we included individual words, two-word phrases, and three-word phrases in our topic model.

3 Results

3.1 Term Frequency

The word cloud in Figure 1 displays the 167 words that were used at least 50 times in the descriptions of the 1,276 courses after removing stopwords. Not surprisingly, the most widely used word was “data”, for a total of 2,322 times. The words “analysis”, “model”, “method”, “learning”, and “system” were each used more than 500 times. Other commonly used words include “algorithm”,

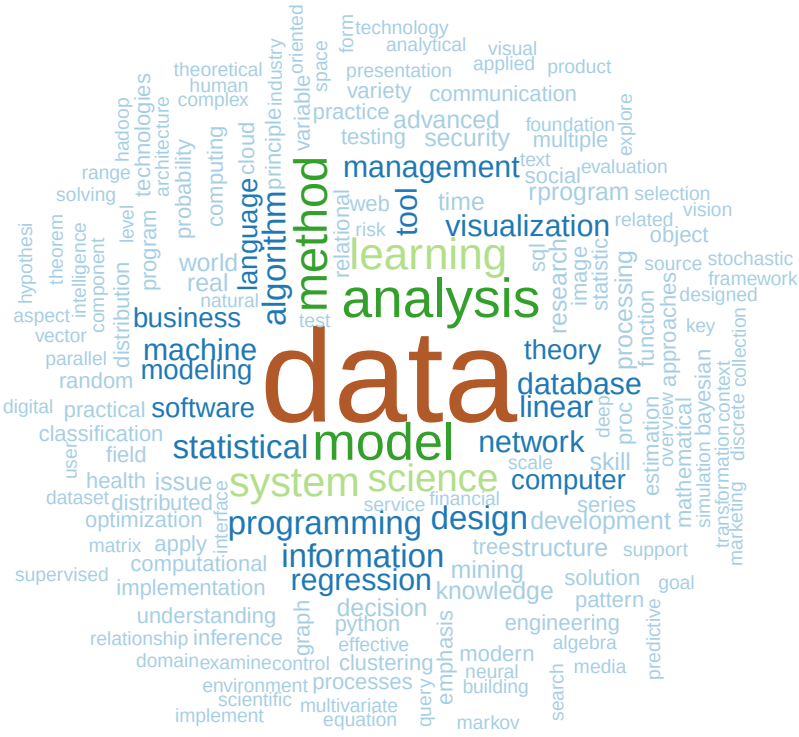


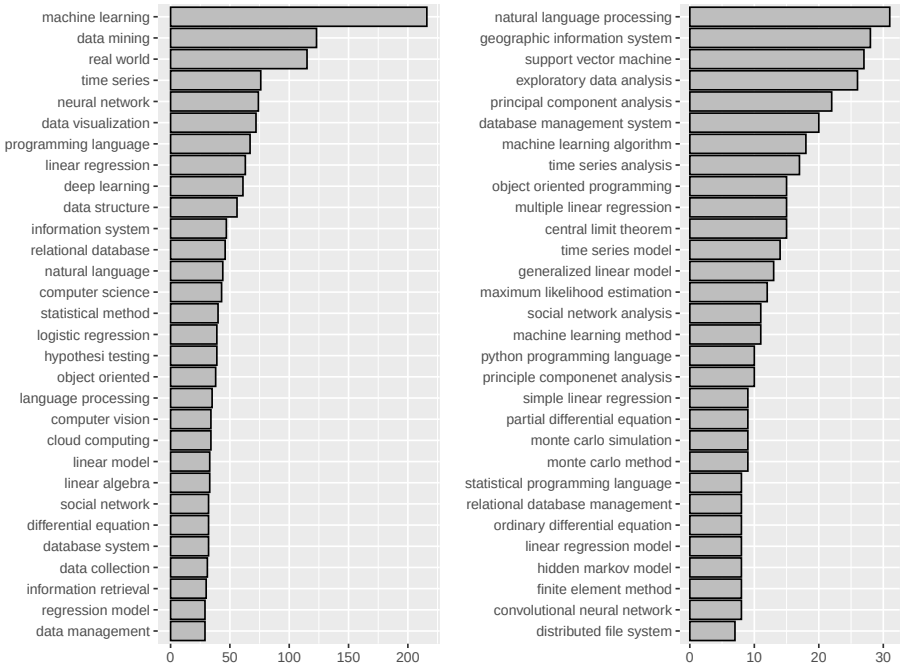
Figure 1: Frequency of individual words used in all course descriptions

"regression", "modeling", "programming", and "visualization". From the frequency analysis, we can see the basic focuses of the data science courses in this study were data analysis, modeling, and data visualization.

Figure 2 shows the top 30 most frequently used two-word phrases and three-word phrases in all course descriptions. Each two-word phrase was used at least 29 times and each three-word phrase was used at least 7 times. These phrases are generally names of analytical models and methods such as "machine learning", "data mining", "linear regression", "natural language processing", and "principal component analysis". We also looked into four-word phrases but only found several to be informative including "markov chain monte carlo" (8 times), "relational database management system" (8 times), "unsupervised machine learning algorithm" (3 times), and "multivariate time series analysis " (3 times).

3.2 Topic Modeling

To determine the number of topics, we first conducted a 5-fold cross-validation. We varied the number of topics from 2 to 25 and calculated the perplexity of the model with a given number of topics. The R package topicmodels (Grün &



(a) Two-word phrases (b) Three-word phrases

Figure 2: Most frequently used phrases

Hornik, 2011) we used for LDA estimation was sensitive to the seed used for the Gibbs sampling algorithm. Therefore, we tried 100 different seeds to get 100 sets of results and then evaluated each set of result to get the best number of topics. Figure 3 shows the perplexities of a topic model with different numbers of topics based on one seed. From it, the model with 14 topics had the smallest perplexity and the perplexity seemed to flatten out after 14 topics. For this particular seed, we would conclude that the best model was the one with 14 topics. Using the perplexity plot, we identified the number of topics for all 100 sets of analyses. Among the 100 sets of analyses, the models with 11, 12, 13, 14, 15, 16, 17, and 18 topics were best models for 1, 7, 35, 27, 20, 4, 5, and 1 times, respectively, based on the perplexity. Therefore, it suggested a model between 13 to 15 topics was probably the best for the course descriptions. We then fitted the models with 13, 14, and 15 topics and investigated the terms and courses associated with each topic. All considered, we found that the model with 14 topics gave a clear representation of different topics and therefore based our discussion on the model with 14 topics.

To understand what each of the 14 topics represented, we first studied the top 30 words and phrases associated with that topic. The topic words were

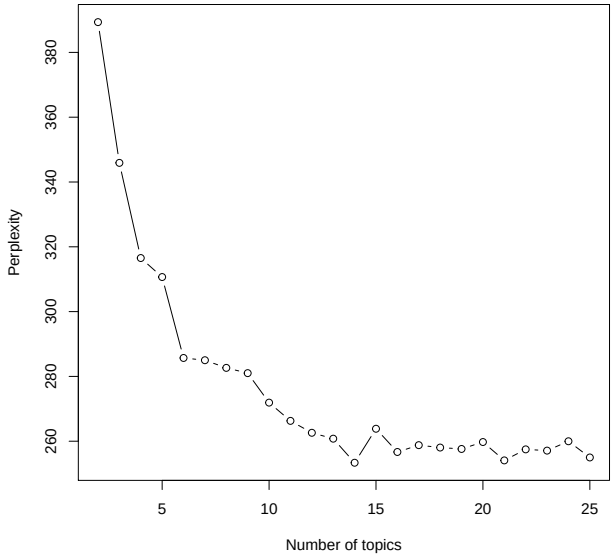


Figure 3: Perplexities of the topic model with different numbers of topics

selected based on the term-scores proposed by Blei and Lafferty (2009). Then, we assigned each course to the highest likely topic and looked through the names of the courses. Based on the analysis, we named the 14 topics, each of which can be viewed as a course to be taught in a data science program. We also identified the commonly taught subjects in each topic/course. We now discuss each of the topics in terms of the most frequently used words and phrases. We will also provide four example classes on each topic from the data science programs analyzed in our study.

Topic 1. Ethics, Privacy, and Security. The first topic is related to research ethics, privacy issues, and data security.[5] The top relevant words and phrases associated with the topic include information, management, security, system, technology, collection, risk, information system, policy, privacy, spatial, ethical, law, ethic, data collection, geographic information system, data management, change, impact, individual, market, access, cost, technical, environment, managing, operation, quantitative, internet, and document. A course in this topic can discuss subjects such as ethics and policy in data analysis, information policy and ethics, data privacy and security, particularly in security and governance of big data, and cyber data security and policy. Example classes include *Behind the*

[5] Note that the order of the named topics might not be the same as the output of the topic modeling in R and does not reflect the relative importance of the topics.

Data: Humans and Values, *Ethics of Big Data*, *Cyber Security Law & Policy*, and *Ethics, Privacy, Security and Governance of Big Data*.

Topic 2. Database Structure and Database Management. The second topic is on database/data structure and database/data management. The top relevant words and phrases associated with the topic include database, system, relational, sql, distributed, parallel, query, architecture, hadoop, relational database, processing, structured, database system, management, mapreduce, memory, transaction, unstructured, storage, query language, database design, management system, database management, file, warehousing, database management system, physical, unstructured data, managing, and selected. A course in this topic can discuss subjects such as different types of database systems, different types of data, database processing and information retrieval, database management systems, big data, and data warehousing. Example classes include *Big Data and NoSQL Program*, *Large-Scale Database Systems*, *Principles of Database Management Systems*, and *Databases and Data Management*.

Topic 3. Data Visualization. The third topic is mainly about visualization, graphical display of data, and exploratory data analysis. The top relevant words and phrases associated with the topic include visualization, tool, principle, communication, data visualization, effective, explore, visual, exploratory, graphic, interactive, insight, exploratory data, exploratory data analysis, critical, perception, technical, apply, aspect, dataset, biology, goal, complex, driven, finding, human, trend, quantitative, environment, and graphical. A course in this topic can discuss subjects such as data visualization techniques and tools, data preparation and preprocessing methods, and types of statistical graphs. Example classes include *Data Visualization*, *Information Visualization and Infographics*, *Visualization of Complex Data*, and *Data Presentation and Visualization with R*.

Topic 4. Algebra. The fourth topic mainly concerns algebra and optimization methods. The top relevant words and phrases associated with the topic include linear, system, function, space, component, matrix, transformation, vector, form, algebra, decomposition, map, reduction, properties, element, linear algebra, spectral, principal, rprogram, dimensional, standard, clustering, cross, dimension, computation, finding, theoretical, equation, primary, principal component analysis. A course in this topic can discuss subjects such as linear and matrix algebra, and numerical methods. Example classes include *Numerical Linear Algebra*, *Computational Algebra*, *Linear Programming*, and *Matrix Algorithms for Data Science*.

Topic 5. Mathematical Foundations and Modeling. This topic is on foundation of mathematics and mathematical modeling. The top relevant words and phrases associated with the topic include theory, mathematical, optimization,

processes, simulation, financial, discrete, stochastic, finance, economic, engineering, modeling, equation, numerical, differential, integration, procedure, differential equation, classical, operation, calculus, transform, continuous, generation, complexity, dynamic, function, complex, control, and matlab. A class on this topic would focus on basic knowledge and foundations of mathematics, optimization methods, and mathematical modeling. Example courses include *Fundamentals of Computational Mathematics*, *Mathematical Modeling*, *Mathematics for Data Scientists*, and *Simulation & Optimization*.

Topic 6. Probability Theory and Statistical Inference. This topic is about basic probability theory and statistical inference. The top relevant words and phrases associated with the topic include statistical, statistic, distribution, estimation, probability, inference, testing, random, bayesian, hypothesis, sampling, variance, sample, hypothesis testing, variable, likelihood, interval, maximum, conditional, parameter, nonparametric, bayes, maximum likelihood, measure, statistical method, limit, prior, statistical inference, confidence, and statistical analysis. A course in this topic would focus on traditional probability and inference topics such as different types of distributions, random variables, sampling distributions, hypothesis testing, and maximum likelihood method. Example classes include *Mathematical Statistics*, *Probability and Statistics for Data Science*, *Bayesian Statistics*, and *Statistical Inference for Data Science*.

Topic 7. Statistical Models. This topic focuses on different types of statistical models for data analysis. The top relevant words and phrases associated with the topic include model, regression, time series, multiple, linear regression, selection, linear, variable, simple, statistical, logistic, forecasting, logistic regression, linear model, parametric, response, regression model, factor, generalized, experimental, interpretation, time series analysis, hierarchical, modeling, multiple linear regression, comparison, sequence, nonlinear, statistical method, and classical. A course in this topic would discuss different types of statistical models such as linear and generalized linear models. Example classes include *Linear Models for Data Science*, *Multivariate Data Analysis*, *Applied Regression Analysis*, and *Experimental Design*.

Topic 8. Statistical Software and Programming. This topic is related to statistical software and basic programming for data analysis. The top relevant words and phrases associated with the topic include programming, algorithm, structure, graph, python, programming language, data structure, tree, matching, flow, efficient, dynamic, complexity, sequence, sorting, object oriented programming, matlab, framework, algorithmic, driven, advanced, code, operation, dataset, package, internet, ethical, measurement, program, and single. A course in this topic could teach how to use software such as R, MATLAB, and Python for data analysis, software programming, and data computing. Example classes include *Statistical Programming in R*, *Systems and Technologies: Python*, *Python for Data Analysis*, and *SAS Programming*.

Topic 9. Machine Learning and Deep Learning This topic is about machine learning and deep learning methods and techniques. The top relevant words and phrases associated with the topic include learning, machine, machine learning, deep, neural network, neural, deep learning, supervised, unsupervised, classification, clustering, tree, artificial, support vector, unsupervised learning, support vector machine, learning algorithm, reinforcement learning, feature, graphical, reinforcement, learning method, decision tree, supervised learning, training, support, mean, dimensionality, machine learning algorithm, and recognition. A course on this topic would introduce different machine learning and deep learning methods and techniques. Example classes include *Neural Networks and Deep Learning*, *(Applied) Machine Learning*, *Machine Learning and Big Data*, and *Deep Learning*.

Topic 10. Business Analytics and Data Mining. This topic is about data mining and business intelligence techniques and methods. The top relevant words and phrases associated with the topic include business, decision, mining, data mining, modeling, intelligence, pattern, predictive, classification, marketing, support, prediction, discovery, identify, association, customer, domain, bioinformatics, tool, healthcare, clustering, organizational, implementing, organization, enterprise, life, topic, descriptive, implement, and exploration. Such a course would be different from a course on machine learning and deep learning in terms of the subjects taught. Example classes include *Data Mining*, *Financial Data Mining*, *Business Analytics and Data Mining*, and *Business Analytics Fundamentals*.

Topic 11. Network Analysis and Text Mining. This topic is about network analysis and text mining/natural language processing. The top relevant words and phrases associated with the topic include network, language, social, web, text, natural, processing, human, search, media, retrieval, natural language, interaction, relationship, social network, natural language processing, language processing, information retrieval, topic, social media, document, probabilistic, indexing, extraction, graph, measure, standard, business, algorithm, and generation. A course on this topic can teach different types of network models, text mining, and graph theory. Example classes include *NLP: Computational Models of Social Meaning*, *Natural Language Processing*, *Text Mining*, and *Social Network Analysis*.

Topic 12. Cloud Computing and Big Data Analysis. This topic is on computing in the cloud and analysis of big data. The top relevant words and phrases associated with the topic include real, computing, world, program, cloud, rprogram, practical, industry, technologies, apply, scale, platform, cloud computing, real world data, dataset, life, aspect, framework, infrastructure, manipulation, cluster, language, cleaning, storage, computation, experimental, survey, internet, statistical method, and quantitative. A class on this topic would focus on how to conduct cloud computing and how to mining data in the cloud. Example courses

include *Cloud Computing*, *Big Data Technologies*, *Big Data Analysis*, and *Cloud Computing for Data Analytics*.

Topic 13. Software Design and Software Engineering. This topic is about software design and software engineering. The top relevant words and phrases associated with the topic include design, software, advanced, implementation, object, user, oriented, control, interface, implement, engineering, common, level, survey, art, software development, package, effect, quality, cycle, code, libraries, experiment, strategies, environment, access, model, exploration, relationship, and real. A course on this topic can teach how to design and develop software, software environment and fundamentals of programming. Example courses include *Programming for Data Science*, *Software Engineering*, *Introduction to Software Development*, and *Computer Systems Programming*.

Topic 14. Applications. This topic is related to the application of data science in different disciplines particularly heath. A notable area is computer vision and image processing. The top relevant words and phrases associated with the topic include computer, computational, image, health, field, foundation, vision, digital, processing, computer science, detection, public, level, goal, medical, theoretical, biological, domain, object, recognition, limited, measurement, extraction, quantitative, interpretation, discipline, feature, organization, filtering, and interpret. A course on this topic may focus on a particular area of applications. Example courses include *Computer Vision*, *Health Data Science*, *Introduction to Biomedical Informatics*, and *Genomics Analytics*.

4 Discussion

Through the analysis of the descriptions of more than 1,200 courses from 56 data science programs offered in the United States, we identified 14 topics or themes that are common in data science training. They are Ethics, Privacy, and Security, Database Structure and Database Management, Data Visualization, Algebra, Mathematical Foundations and Modeling, Probability Theory and Statistical Inference, Statistical Models, Statistical Software and Programming, Machine Learning and Deep Learning, Business Analytics and Data Mining, Network Analysis and Text Mining, Cloud Computing and Big Data Analysis, Software Design and Software Engineering, and Applications. All 14 topics contributed about equally to the contents of all the courses analyzed in the study, with Probability Theory and Statistical Inference contributing the most, 7.39% and Algebra the least, 6.94%.

Data science training or even a single course is often an integrated unit. Therefore, the 14 topics are more or less related and can share the same contents, as reflected in terms associated with the topics. For example, when teaching the discipline-specific applications, it cannot be avoided to discuss data mining and machine learning techniques, data visualization, and data management to

demonstrate their utilization. The 14-topic model in our study includes the following two topics – "Business Analytics and Data Mining" and "Machine Learning and Deep Learning". Although the two topics shared some same subjects, Business Analytics and Data Mining seemed to focus more on traditional big data techniques often developed in the statistics discipline such as classification and regression tree, mixture model, and discriminant analysis as well as business intelligence. Machine Learning and Deep Learning, on the other hand, covered more topics developed in the computer science discipline such as different types of learning methods, neural network, pattern recognition, and support vector machine techniques. Similarly, we identified a topic on Statistical Software and Programming as well as a topic on Software Design and Software Engineering. The former focused more on the use of software such as R, Python, and MATLAB for practical data analysis and the latter concerned more about software development.

Although fourteen topics provided the best result for our topic model based on cross-validation, the fourteen topics did not necessarily cover all the topics offered in all data science programs analyzed in the study. For example, in the process of understanding the meaning of each topic, we found that Computer Vision stood out as an important topic taught in the data science programs. In addition, some of the topics might be split into multiple topics. For example, the topic of Business Analytics and Data Mining can be split into two. Network Analysis and Text Mining can also be viewed as two separated but closely related topics.

Among the fourteen topics, Algebra, Mathematical Foundations and Modeling, Probability Theory and Statistical Inference, Statistical Models, and Statistical Software and Programming are arguably the traditionally strong areas of the discipline of statistics. Database Structure and Database Management, Machine Learning and Deep Learning, Network Analysis and Text Mining, Cloud Computing and Big Data Analysis, and Software Design and Software Engineering can be viewed as emerging and important areas in the discipline of computer science. Data Visualization and Data Mining have been the focuses of both statistics and computer science disciplines. Ethics, Privacy, and Security is becoming an important topic in both disciplines. The fourteen topics together speak unequivocally that data science is an interdisciplinary area that integrates statistics, computer science, and substantive fields (Applications).

We have chosen to focus on fourteen topics in the analysis. If only thirteen topics were kept, the topic "Software Design and Software Engineering" would drop out. On the other hand, if fifteen topics were used, then "Network Analysis" and "Text Mining" can be broken into two topics.

Although we arrived at the identified topics through empirical analysis of course descriptions, these topics aligned well with the existing literature. Particularly, they were consistent with the Data Science Knowledge Framework by Fayyad and Hamutcu (2020). Our results also reflected the finding by Gorman and Klimberg (2014). In their study, they analyzed the curriculum of 17 business analytics programs and interviewed 11 programs. The 14 subjects that

they identified, such as Introduction to Statistics, Regression, Multivariate, and Time Series Analysis, seemed to mostly fall within the scope of applied statistics. However, they also pointed out three trending developments including Big Data: Internet of Things, Unstructured Data and Semantic Analysis, and Network Analytics. In another study, Song and Zhu (2016) investigated both undergraduate (7 in total) and graduate (15 in total) curricula in data sciences and proposed several approaches for data science educations. The topics identified in our study can be combined with their approaches. Overall, our study provides additional empirical support to the existing literature for understanding what is data science.

4.1 Limitations

Our study has several limitations. First, the analysis used data only from the programs with "Data science" in their titles. There are many data analytics and business analytics programs tracked by IAA. In practice, the differences in "Data science" and "Business analytics" might not be large. It can be interesting to see whether the course information from the programs can be combined for more comprehensive data analysis. Second, the findings in this study were based on the analysis of course descriptions from data science programs in academic institutes. However, the results may or may not align with industry/applied/business applications of data science. In the future, the results can be compared to the analysis of other information, such as job postings for data science positions, to identify potential similarities and differences between academic training in data science and data science as practiced in industry. Third, we only considered the data science programs in the US. The findings may not be generalized to other countries.

4.2 Conclusion

The goal of this study is to understand what data science is through the mining of the courses offered by data science programs in the US to hopefully provide a better definition of data science. We adopted a bottom-up approach to mining the description information of individual courses taught in current data sciences programs. Although we identified fourteen topics among all the courses, it is still difficult to provide a concise and conclusive definition of data science. However, we believe our results can provide useful information on how to operate data science programs. The results of our study further reiterate the notion that data science is at the intersection of statistics, computer science, and applications. A major contribution of our study is to provide empirical support to a better understanding of data science.

Acknowledgment

This study is partly supported by a grant from the Department of Education (R305D210023). However, the contents of the study do not necessarily represent

the policy of the Department of Education, and you should not assume endorsement by the Federal Government. We thank Wen Qu and Tyler Wilcox for their helpful comments and suggestions that improved the study.

References

Blei, D. M., & Lafferty, J. D. (2009). Text mining: Classification, clustering, and applications. In A. Srivastava & M. Sahami (Eds.), (pp. 71–93). Chapman & Hall/CRC.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022. doi: https://doi.org/10.5555/944919.944937

Broman, K. W. (2013). *Data science is statistics.* Retrieved from `https://kbroman.wordpress.com/2013/04/05/data-science-is-statistics/` (Retrieved on Mar 12, 2021)

Cegielski, C. G., & Jones-Farmer, L. A. (2016). Knowledge, skills, and abilities for entry-level business analytics positions: A multi-method study. *Decision Sciences Journal of Innovative Education*, *14*(1), 91–118. doi: https://doi.org/10.1111/dsji.12086

Cleveland, W. S. (2001). Data science: an action plan for expanding the technical areas of the field of statistics. *International statistical review*, *69*(1), 21–26. doi: https://doi.org/10.1002/sam.11239

Dhar, V. (2013). Data science and prediction. *Communications of the ACM*, *56*(12), 64–73. doi: https://doi.org/10.1145/2500499

Donoho, D. (2017). 50 years of data science. *Journal of Computational and Graphical Statistics*, *26*(4), 745-766. doi: https://doi.org/10.1080/10618600.2017.1384734

Fayyad, U., & Hamutcu, H. (2020). Toward foundations for data science and analytics: A knowledge framework for professional standards. *Havard Data Science Review*, *2*, 2. doi: https://doi.org/10.1162/99608f92.1a99e67a

Gorman, M. F., & Klimberg, R. K. (2014). Benchmarking academic programs in business analytics. *Interfaces*, *44*(3), 329–341. doi: https://doi.org/10.1287/inte.2014.0739

Grün, B., & Hornik, K. (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, *40*(13), 1–30. doi: https://doi.org/10.18637/jss.v040.i13

Hayashi, C. (1998). What is data science? fundamental concepts and a heuristic example. In *Data science, classification, and related methods* (pp. 40–51). Springer.

Hey, T., Tansley, S., & Tolle, K. (2009). *The fourth paradigm: data-intensive scientific discovery* (Vol. 1). Microsoft research Redmond, WA.

Hickman, L., Thapa, S., Tay, L., Cao, M., & Srinivasan, P. (2020). Text preprocessing for text mining in organizational research: Review and recommendations. *Organizational Research Methods*, 1-33. doi: https://doi.org/10.1177/1094428120971683

Lau, J. H., Baldwin, T., & Newman, D. (2013). On collocations and topic models. *ACM Transactions on Speech and Language Processing (TSLP)*, *10*(3), 1–14. doi: https://doi.org/10.1145/2483969.2483972

Naur, P. (1966, July). The science of datalogy. *Communications of the ACM*, *9*(7), 485. doi: https://doi.org/10.1145/365719.366510

Naur, P. (1974). *Concise survey of computer methods.* Petrocelli Books.

Nokel, M., & Loukachevitch, N. (2015). A method of accounting bigrams in topic models. In *Proceedings of the 11th workshop on multiword expressions* (pp. 1–9).

Silver, N. (2013). *What i need from statisticians.* Retrieved from `https://www.statisticsviews.com/article/nate-silver-what-i-need-from-statisticians/` (Retrieved on 3/12/2021)

Song, I.-Y., & Zhu, Y. (2016). Big data and data science: what should we teach? *Expert Systems*, *33*(4), 364–373. doi: https://doi.org/10.1111/exsy.12130

Tukey, J. W. (1962). The future of data analysis. *The annals of mathematical statistics*, *33*(1), 1–67. doi: https://doi.org/10.1214/aoms/1177704711

Van Dyk, D., Fuentes, M., Jordan, M. I., Newton, M., Ray, B. K., Lang, D. T., & Wickham, H. (2015). *Asa statement on the role of statistics in data science.* AMSTAT News. Retrieved from `https://magazine.amstat.org/blog/2015/10/01/asa-statement-on-the-role-of-statistics-in-data-science/`

Vijayarani, S., Ilamathi, M. J., Nithya, M., et al. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

Journal of Behavioral Data Science, 2021, *1* (1), 17–33.
DOI: https://doi.org/10.35566/jbds/v1n1/p2

Moments Calculation for the Doubly Truncated Multivariate Normal Density

B. G. Manjunath[1] and Stefan Wilhelm[2]

[1] School of Mathematics and Statistics,
University of Hyderabad, Hyderabad, India.
bgmanjunath@gmail.com
[2] Department of Finance, University of Basel, Switzerland
Stefan.Wilhelm@financial.com

Abstract. In the present article, we derive an explicit expression for the truncated mean and variance for the multivariate normal distribution with arbitrary rectangular double truncation. We use the moment generating approach of Tallis (1961) and extend it to general $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and all combinations of truncation. As part of the solution, we also give a formula for the bivariate marginal density of truncated multinormal variates. We also prove an invariance property of some elements of the inverse covariance after truncation. Computer algorithms for computing the truncated mean, variance and the bivariate marginal probabilities for doubly truncated multivariate normal variates have been written in R and are presented along with three examples.

Keywords: Multivariate normal · Double truncation · Moment generating function · Bivariate marginal density function · Graphical models · Conditional independence

1 Introduction

The multivariate normal distribution arises frequently and has a wide range of applications in fields such as multivariate regression, Bayesian statistics and the analysis of Brownian motion. One motivation to deal with moments of the truncated multivariate normal distribution comes from the analysis of special financial derivatives ("auto-callables" or "Expresszertifikate") in Germany. These products can expire early depending on some restrictions of the underlying trajectory, if the underlying value is above or below certain call levels. In the framework of Brownian motion the finite-dimensional distributions for log returns at any d points in time are multivariate normal. When some of the multinormal variates $\mathbf{X} = (x_1, \ldots, x_d)' \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are subject to inequality constraints (e.g. $a_i \leq x_i \leq b_i$), this results in truncated multivariate normal distributions.

Several types of truncations and their moment calculation have been described so far, for example the one-sided rectangular truncation $\boldsymbol{x} \geq \boldsymbol{a}$ (Tallis,

1961), the rather unusual elliptical and radial truncations $\boldsymbol{a} \leq \boldsymbol{x}'\mathbf{R}\boldsymbol{x} \leq \boldsymbol{b}$ (Tallis, 1963) and the plane truncation $\mathbf{C}\boldsymbol{x} \geq \boldsymbol{p}$ (Tallis, 1965). Linear constraints like $\boldsymbol{a} \leq \mathbf{C}\boldsymbol{x} \leq \boldsymbol{b}$ can often be reduced to rectangular truncation by transformation of the variables (in case of a full rank matrix $\mathbf{C} : \boldsymbol{a}^* = \mathbf{C}^{-1}\boldsymbol{a} \leq \boldsymbol{x} \leq \mathbf{C}^{-1}\boldsymbol{b} = \boldsymbol{b}^*$), which makes the double rectangular truncation $\boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}$ especially important.

The existing works on moment calculations differ in the number of variables they consider (univariate, bivariate, multivariate) and the types of rectangular truncation they allow (single vs. double truncation). Single or one-sided truncation can be either from above ($\boldsymbol{x} \leq \boldsymbol{a}$) or below ($\boldsymbol{x} \geq \boldsymbol{a}$), but only on one side for all variables, whereas double truncation $\boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}$ can have both lower and upper truncation points. Other distinguishing features of previous works are further limitations or restrictions they impose on the type of distribution (e.g. zero mean) and the methods they use to derive the results (e.g. direct integration or moment-generating function).

Rosenbaum (1961) gave an explicit formula for the moments of the bivariate case with single truncation from below in both variables by direct integration. His results for the bivariate normal distribution have been extended by Shah and Parikh (1964), Regier and Hamdan (1971) and Muthén (1990) to double truncation.

For the multivariate case, Tallis (1961) derived an explicit expression for the first two moments in case of a singly truncated multivariate normal density with zero mean vector and the correlation matrix $\mathbf{R}$ using the moment generating function. Amemiya (1974) and Lee (1979) extended the Tallis (1961) derivation to a general covariance matrix $\boldsymbol{\Sigma}$ and also evaluated the relationship between the first two moments. Gupta and Tracy (1976) and Lee (1983) gave very simple recursive relationships between moments of any order for the doubly truncated case. However, except for the mean, there are fewer equations than parameters. Therefore, these recurrent conditions do not uniquely identify moments of order ≥ 2 and are not sufficient for the computation of the variance and other higher order moments.

Table 1 summarizes our survey of existing publications dealing with the computation of truncated moments and their limitations. Even though the rectangular truncation $\boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}$ can be found in many situations, no explicit moment formulas for the truncated mean and variance in the general multivariate case of double truncation from below and/or above have been presented so far in the literature and are readily apparent. The contribution of this paper is to derive these formulas for the first two truncated moments and to extend and generalize existing results on moment calculations from especially Tallis (1961), Lee (1983), Leppard and Tallis (1989), and Muthén (1990). Besides, we also refer Kan and Robotti (2017) and Arismendi (2013) for the moment computation of folded and truncated multivariate normal distribution. However, for moments computation for skewed and extended skew normal distribution, we refer Kan and Robotti (2017) and Arellano-Valle and Genton (2005). In the sequel, we also make a note on the existing R package "MomTrunc" (see Galarza C.E. & V.H., 2021) for numerical computation of moments of folded and truncated multivariate normal

Table 1. Survey of previous works on the moments for the truncated multivariate normal distribution

Author	#Variates	Truncation	Focus
Rosenbaum (1961)	bivariate	single	moments for bivariate normal variates with single truncation, $b_1 < y_1 < \infty, b_2 < y_2 < \infty$
Tallis (1961)	multivariate	single	moments for multivariate normal variates with single truncation from below
Shah and Parikh (1964)	bivariate	double	recurrence relations between moments
Regier and Hamdan (1971)	bivariate	double	an explicit formula only for the case of truncation from below at the same point in both variables
Amemiya (1974)	multivariate	single	relationship between first and second moments
Gupta and Tracy (1976)	multivariate	double	recurrence relations between moments
Lee (1979)	multivariate	single	recurrence relations between moments
Lee (1983)	multivariate	double	recurrence relations between moments
Leppard and Tallis (1989)	multivariate	single	moments for multivariate normal distribution with single truncation
Muthén (1990)	bivariate	double	moments for bivariate normal distribution with double truncation, $b_1 < y_1 < a_1, b_2 < y_2 < a_2$
Manjunath and Wilhelm	multivariate	double	moments for multivariate normal distribution with double truncation in all variables $\boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}$

and Student's t-distribution. However, the aforementioned package also suggests the "tmvtnorm" package (e.g., Wilhelm & Manjunath, 2012), which is solely based on the results presented in this note. Finally, we also refer Genz (1992) for the numerical computation for the multivariate normal probabilities.

The rest of this paper is organized as follows. Section 2 presents the moment generating function (m.g.f) for the doubly truncated multivariate normal case. In Section 3, we derive the first and second moments by differentiating the m.g.f. These results are completed in Section 4 by giving a formula for computing the bivariate marginal density. In Section 5, we present two numerical examples and compare our results with simulation results. Section 6 links our results to the theory of graphical models and derives some properties of the inverse covariance matrix. Finally, Section 7 summarizes our results and gives an outlook for further research.

2 Moment Generating Function

The d–dimensional normal density with location parameter vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and non-singular covariance matrix $\boldsymbol{\Sigma}$ is given by

$$\varphi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)' \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right\}, \ \boldsymbol{x} \in \mathbb{R}^d. \quad (1)$$

The pertaining distribution function is denoted by $\Phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x})$. Correspondingly, the multivariate truncated normal density, truncated at $\boldsymbol{a}$ and $\boldsymbol{b}$, in $\mathbb{R}^d$, is defined as

$$\varphi_{\alpha\,\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}) = \begin{cases} \frac{\varphi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x})}{P\{\boldsymbol{a}\leq\mathbf{X}\leq\boldsymbol{b}\}}, & \text{for } \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Denote $\alpha = P\left\{\boldsymbol{a} \leq \mathbf{X} \leq \boldsymbol{b}\right\}$ as the fraction after truncation.

The moment generating function of a d–dimensional truncated random variable $\mathbf{X}$, truncated at $\boldsymbol{a}$ and $\boldsymbol{b}$, in $\mathbb{R}^d$, having the density $f(\boldsymbol{x})$ is defined as the d–fold integral of the form

$$m(\boldsymbol{t}) = E\left(e^{\boldsymbol{t}'\mathbf{X}}\right) = \int_{\boldsymbol{a}}^{\boldsymbol{b}} e^{\boldsymbol{t}'\boldsymbol{x}} f(\boldsymbol{x}) d\boldsymbol{x}.$$

Therefore, the m.g.f for the density in (2) is

$$m(\boldsymbol{t}) = \frac{1}{\alpha(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \int_{\boldsymbol{a}}^{\boldsymbol{b}} \exp\left\{-\frac{1}{2}\left[\left(\boldsymbol{x}-\boldsymbol{\mu}\right)' \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right) - 2\boldsymbol{t}'\boldsymbol{x}\right]\right\} d\boldsymbol{x}. \quad (3)$$

In the following, the moments are first derived for the special case $\boldsymbol{\mu} = \mathbf{0}$. Later, the results will be generalized to all $\boldsymbol{\mu}$ by applying a location transformation.

Now, consider only the exponent term in 3 for the case $\boldsymbol{\mu} = \mathbf{0}$. Then we have

$$-\frac{1}{2}\left[\boldsymbol{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - 2\boldsymbol{t}'\boldsymbol{x}\right]$$

which can also be written as

$$\frac{1}{2}\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t} - \frac{1}{2}\left[\left(\boldsymbol{x}-\boldsymbol{\xi}\right)' \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\xi}\right)\right],$$

where $\boldsymbol{\xi} = \boldsymbol{\Sigma}\boldsymbol{t}$.

Consequently, the m.g.f of the rectangularly doubly truncated multivariate normal is

$$m(\boldsymbol{t}) = \frac{e^T}{\alpha(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \int_{\boldsymbol{a}}^{\boldsymbol{b}} \exp\left\{-\frac{1}{2}\left[\left(\boldsymbol{x}-\boldsymbol{\xi}\right)' \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\xi}\right)\right]\right\} d\boldsymbol{x}, \quad (4)$$

where $T = \frac{1}{2}\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t}$.

The above equation can be further reduced to

$$m(\boldsymbol{t}) = \frac{e^T}{\alpha(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \int_{\boldsymbol{a}-\boldsymbol{\xi}}^{\boldsymbol{b}-\boldsymbol{\xi}} \exp\left\{-\frac{1}{2}\boldsymbol{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{x}\right\} d\boldsymbol{x}. \tag{5}$$

For notational convenience, we write equation 5 as

$$m(\boldsymbol{t}) = e^T \Phi_{\alpha\boldsymbol{\Sigma}} \tag{6}$$

where

$$\Phi_{\alpha\boldsymbol{\Sigma}} = \frac{1}{\alpha(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \int_{\boldsymbol{a}-\boldsymbol{\xi}}^{\boldsymbol{b}-\boldsymbol{\xi}} \exp\left\{-\frac{1}{2}\boldsymbol{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{x}\right\} d\boldsymbol{x}.$$

3 First and Second Moment Calculation

In this section, we derive the first and second moments of the rectangularly doubly truncated multivariate normal density by differentiating the m.g.f..

Consequently, by taking the partial derivative of (6) with respect to t_i we have

$$\frac{\partial m(\boldsymbol{t})}{\partial t_i} = e^T \frac{\partial \Phi_{\alpha\boldsymbol{\Sigma}}}{\partial t_i} + \Phi_{\alpha\boldsymbol{\Sigma}} \frac{\partial e^T}{\partial t_i}. \tag{7}$$

In the above equation the only essential terms that will be simplified are

$$\frac{\partial e^T}{\partial t_i} = e^T \sum_{k=1}^{d} \sigma_{i,k} t_k$$

and

$$\frac{\partial \Phi_{\alpha\boldsymbol{\Sigma}}}{\partial t_i} = \frac{\partial}{\partial t_i} \int_{a_1^*}^{b_1^*} \dots \int_{a_d^*}^{b_d^*} \varphi_{\alpha\boldsymbol{\Sigma}}(\boldsymbol{x}) dx_d \dots dx_1, \tag{8}$$

where $a_i^* = a_i - \sum_{k=1}^{d} \sigma_{i,k} t_k$ and $b_i^* = b_i - \sum_{k=1}^{d} \sigma_{i,k} t_k$. Subsequently, (8) is

$$\frac{\partial \Phi_{\alpha\boldsymbol{\Sigma}}}{\partial t_i} = \sum_{k=1}^{d} \sigma_{i,k} \left(F_k(a_k^*) - F_k(b_k^*)\right), \tag{9}$$

where

$$F_i(x) = \int_{a_1^*}^{b_1^*} \dots \int_{a_{i-1}^*}^{b_{i-1}^*} \int_{a_{i+1}^*}^{b_{i+1}^*} \dots \int_{a_d^*}^{b_d^*} \varphi_{\alpha\boldsymbol{\Sigma}}(x_1, .., x_{i-1}, x, x_{i+1}, ..x_d) dx_d \dots dx_{i+1} dx_{i-1} \dots dx_1. \tag{10}$$

Note that at $t_k = 0$, for all $k = 1, 2, ..., d$, we have $a_i^* = a_i$ and $b_i^* = b_i$. Therefore, $F_i(x)$ will be the i–th marginal density. An especially convenient way of computing these one-dimensional marginals is given in Cartinhour (1990).

From (7) – (9) for $k = 1, 2, ..., d$ all $t_k = 0$. Hence, the first moment is

$$E(X_i) = \frac{\partial m(\boldsymbol{t})}{\partial t_i}|_{\boldsymbol{t}=\mathbf{0}} = \sum_{k=1}^{d} \sigma_{i,k} \left(F_k(a_k) - F_k(b_k) \right). \tag{11}$$

Now, by taking the partial derivative of (7) with respect to t_j, we have

$$\frac{\partial^2 m(\boldsymbol{t})}{\partial t_j \partial t_i} = e^T \frac{\partial^2 \Phi_{\alpha\boldsymbol{\Sigma}}}{\partial t_j \partial t_i} + \frac{\partial \Phi_{\alpha\boldsymbol{\Sigma}}}{\partial t_i} \frac{\partial e^T}{\partial t_j} + \Phi_{\alpha\boldsymbol{\Sigma}} \frac{\partial^2 e^T}{\partial t_j \partial t_i} + \frac{\partial e^T}{\partial t_i} \frac{\partial \Phi_{\alpha\boldsymbol{\Sigma}}}{\partial t_j}. \tag{12}$$

The essential terms for simplification are

$$\frac{\partial^2 e^T}{\partial t_j \partial t_i} = \sigma_{i,j}$$

and clearly, the partial derivative of 9 with respect to t_j gives

$$\frac{\partial^2 \Phi_{\alpha\boldsymbol{\Sigma}}}{\partial t_j \partial t_i} = \sum_{k=1}^{d} \left(\sigma_{i,k} \frac{\partial F_k(a_k^*)}{\partial t_j} \right) - \sum_{k=1}^{d} \left(\sigma_{i,k} \frac{\partial F_k(b_k^*)}{\partial t_j} \right). \tag{13}$$

In the above equation, merely consider the partial derivative of the marginal density $F_k(a_k^*)$ with respect to t_j. With further simplification, it reduces to

$$\begin{aligned} \frac{\partial F_k(a_k^*)}{\partial t_j} &= \frac{\partial}{\partial t_j} \int_{a_1^*}^{b_1^*} ... \int_{a_{k-1}^*}^{b_{k-1}^*} \int_{a_{k+1}^*}^{b_{k+1}^*} ... \int_{a_d^*}^{b_d^*} \varphi_{\alpha\boldsymbol{\Sigma}}(x_1, .., x_{k-1}, a_k^*, x_{k+1}, ..x_d) d\boldsymbol{x}_{-k} \\ &= \frac{\sigma_{j,k} a_k^* F_k(a_k^*)}{\sigma_{k,k}} \\ &\quad + \sum_{q \neq k} \left(\sigma_{j,q} - \frac{\sigma_{k,q}\sigma_{j,k}}{\sigma_{k,k}} \right) \left(F_{k,q}(a_k^*, a_q^*) - F_{k,q}(a_k^*, b_q^*) \right), \end{aligned} \tag{14}$$

where

$$\begin{aligned} &F_{k,q}(x, y) = \\ &\int_{a_1^*}^{b_1^*} ... \int_{a_{k-1}^*}^{b_{k-1}^*} \int_{a_{k+1}^*}^{b_{k+1}^*} ... \int_{a_{q-1}^*}^{b_{q-1}^*} \int_{a_{q+1}^*}^{b_{q+1}^*} ... \int_{a_d^*}^{b_d^*} \varphi_{\alpha\boldsymbol{\Sigma}}(x, y, \boldsymbol{x}_{-k,-q}) d\boldsymbol{x}_{-k,-q}, \end{aligned} \tag{15}$$

and the short form $\boldsymbol{x}_{-k}$ denotes the vector $(x_1, .., x_{k-1}, x_{k+1}, ..x_d)'$ in $(d-1)$–dimensions and $\boldsymbol{x}_{-k,-q}$ denotes the $(d-2)$–dimensional vector $(x_1, ..., x_{k-1}, x_{k+1}, ..., x_{q-1}, x_{q+1}, ..., x_d)'$ for $k \neq q$. The above equation (14) is deduced from Lee (1979, pp. 167). Note that for all $t_k = 0$ the term $F_{k,q}(x, y)$ will be the bivariate marginal density for which we will give a formula in the next section.

Subsequently, $\frac{\partial F_k(b_k^*)}{\partial t_j}$ can be obtained by substituting a_k^* by b_k^*. From 12 – (15) at all $t_k = 0$, $k = 1, 2, ..., d$, the second moment is

$$\begin{aligned} E(X_i X_j) &= \frac{\partial^2 m(\boldsymbol{t})}{\partial t_j \partial t_i}|_{\boldsymbol{t}=\boldsymbol{0}} \\ &= \sigma_{i,j} + \sum_{k=1}^{d} \sigma_{i,k} \frac{\sigma_{j,k} \left(a_k F_k(a_k) - b_k F_k(b_k)\right)}{\sigma_{k,k}} \\ &\quad + \sum_{k=1}^{d} \sigma_{i,k} \sum_{q \neq k} \left(\sigma_{j,q} - \frac{\sigma_{k,q}\sigma_{j,k}}{\sigma_{k,k}}\right) \left[\left(F_{k,q}(a_k, a_q) - F_{k,q}(a_k, b_q)\right)\right. \\ &\quad \left. - \left(F_{k,q}(b_k, a_q) - F_{k,q}(b_k, b_q)\right)\right]. \end{aligned} \tag{16}$$

Having derived expressions for the first and second moments for double truncation in case of $\boldsymbol{\mu} = \boldsymbol{0}$, we will now generalize to all $\boldsymbol{\mu}$. If $\boldsymbol{Y} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{a}^* \leq \boldsymbol{y} \leq \boldsymbol{b}^*$, then $\mathbf{X} = \boldsymbol{Y} - \boldsymbol{\mu} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{a} = \boldsymbol{a}^* - \boldsymbol{\mu} \leq \boldsymbol{x} \leq \boldsymbol{b}^* - \boldsymbol{\mu} = \boldsymbol{b}$ and $E(\boldsymbol{Y}) = E(\mathbf{X}) + \boldsymbol{\mu}$ and $Cov(\boldsymbol{Y}) = Cov(\mathbf{X})$. Equations 11 and 16 can then be used to compute $E(\mathbf{X})$ and $Cov(\mathbf{X})$. Hence, for general $\boldsymbol{\mu}$, the first moment is

$$E(Y_i) = \sum_{k=1}^{d} \sigma_{i,k} \left(F_k(a_k) - F_k(b_k)\right) + \mu_i. \tag{17}$$

The covariance matrix

$$Cov(Y_i, Y_j) = Cov(X_i, X_j) = E(X_i X_j) - E(X_i)E(X_j) \tag{18}$$

is invariant to the shift in location.

The equations 17 and 18 in combination with 11 and 16 form our derived result allow the calculation of the truncated mean and truncated variance for general double truncation. A formula for the term $F_{k,q}(x_k, x_q)$, the bivariate marginal density, will be given in the next section.

We have implemented the moment calculation for mean vector `mean`, covariance matrix `sigma` and truncation vectors `lower` and `upper` as a function `mtmvnorm(mean, sigma, lower, upper)` in the R package `tmvtnorm` (Wilhelm & Manjunath, 2010, 2012), where the code is open source. In Section 5, we will show an example of this function.

4 Bivariate Marginal Density Computation

In order to compute the bivariate marginal density in this section, we follow Tallis (1961, p. 223) and Leppard and Tallis (1989) that implicitly used the bivariate marginal density as part of the moments calculation for single truncation, evaluated at the integration bounds. However, we extend it to the doubly truncated case and state the function for all points within the support region.

Without loss of generality we use a z-transformation for all variates $\boldsymbol{x} = (x_1, \ldots, x_d)'$ as well as for all lower and upper truncation points $\boldsymbol{a} = (a_1, \ldots, a_d)'$ and $\boldsymbol{b} = (b_1, \ldots, b_d)'$, resulting in a $N(0, \mathbf{R})$ distribution with correlation matrix $\mathbf{R}$ for the standardized untruncated variates. In this section we treat all variables as if they are z-transformed, leaving the notation unchanged.

For computing the bivariate marginal density $F_{q,r}(x_q, x_r)$ with $a_q \leq x_q \leq b_q, a_r \leq x_r \leq b_r, q \neq r$, we use the fact that for truncated normal densities the conditional densities are also truncated normal. The following relationship holds for $x_s, z_s \in \mathbb{R}^{d-2}$ conditionally on $x_q = c_q$ and $x_r = c_r$ $(s \neq q \neq r)$:

$$\alpha^{-1}\varphi_d(x_s, x_q = c_q, x_r = c_r; \mathbf{R}) = \alpha^{-1}\varphi(c_q, c_r; \rho_{qr})\varphi_{d-2}(z_s; \mathbf{R}_{qr}), \tag{19}$$

where

$$z_s = (x_s - \beta_{sq.r}c_q - \beta_{sr.q}c_r)/\sqrt{(1-\rho^2_{sq})(1-\rho^2_{sr.q})} \tag{20}$$

and $\mathbf{R}_{qr}$ is the matrix of the second-order partial correlation coefficients for $s \neq q \neq r$. $\beta_{sq.r}$ and $\beta_{sr.q}$ are the partial regression coefficients of x_s on x_q and x_r , respectively, and $\rho_{sr.q}$ is the partial correlation coefficient between x_s and x_r for fixed x_q.

Integrating out $(d-2)$ variables x_s leads to $F_{q,r}(x_q, x_r)$ as a product of a bivariate normal density $\varphi(x_q, x_r)$ and a $(d-2)$-dimension normal integral Φ_{d-2}:

$$\begin{aligned} F_{q,r}(x_q = c_q, x_r = c_r) = &\int_{a_1}^{b_1} \ldots \int_{a_{q-1}}^{b_{q-1}} \int_{a_{q+1}}^{b_{q+1}} \ldots \int_{a_{r-1}}^{b_{r-1}} \\ &\int_{a_{r+1}}^{b_{r+1}} \ldots \int_{a_d}^{b_d} \varphi_{\alpha R}(x_s, c_q, c_r) dx_s \\ &= \alpha^{-1}\varphi(c_q, c_r; \rho_{qr})\Phi_{d-2}(A^q_{rs}; B^q_{rs}; \mathbf{R}_{qr}) \end{aligned} \tag{21}$$

where A^q_{rs} and B^q_{rs} denote the lower and upper integration bounds of Φ_{d-2} given $x_q = c_q$ and $x_r = c_r$:

$$A^q_{rs} = (a_s - \beta_{sq.r}c_q - \beta_{sr.q}c_r)/\sqrt{(1-\rho^2_{sq})(1-\rho^2_{sr.q})} \tag{22}$$

$$B^q_{rs} = (b_s - \beta_{sq.r}c_q - \beta_{sr.q}c_r)/\sqrt{(1-\rho^2_{sq})(1-\rho^2_{sr.q})}. \tag{23}$$

The computation of $F_{q,r}(x_q, x_r)$ just needs the evaluation of the normal integral Φ_{d-2} in $d-2$ dimensions, which is readily available in most statistics software packages, for example, as the function `pmvnorm()` in the R package `mvtnorm` (Genz et al., 2012). The bivariate marginal density function `dtmvnorm(x, mean, sigma, lower, upper, margin=c(q,r))` is also part of the R package `tmvtnorm` (Wilhelm & Manjunath, 2010, 2012), where readers can find the source code as well as help files and additional examples.

5 Numerical Examples

Example 1

We will use the following bivariate example with $\boldsymbol{\mu} = (0.5, 0.5)'$ and covariance matrix $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 1.2 \\ 1.2 & 2 \end{pmatrix}$$

as well as lower and upper truncation points $\boldsymbol{a} = (-1, -\infty)', \boldsymbol{b} = (0.5, 1)'$, i.e. x_1 is doubly, while x_2 is singly truncated. The bivariate marginal density $F_{q,r}(x, y)$ is the density function itself and is shown in figure 1, and the one-dimensional densities $F_k(x)$ $(k = 1, 2)$ are shown in in figure 2.

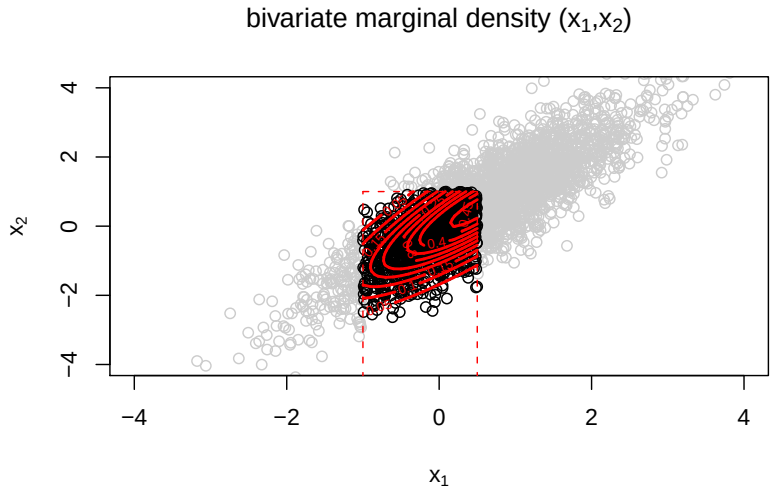


Figure 1. Contour plot for the bivariate truncated density function

The moment calculation for our example can be performed in R as

```
> library(tmvtnorm)
> mu    <- c(0.5, 0.5)
> sigma <- matrix(c(1, 1.2, 1.2, 2), 2, 2)
> a     <- c(-1, -Inf)
> b     <- c(0.5, 1)
> moments <- mtmvnorm(mean=mu, sigma=sigma,
>            lower=a, upper=b)
```

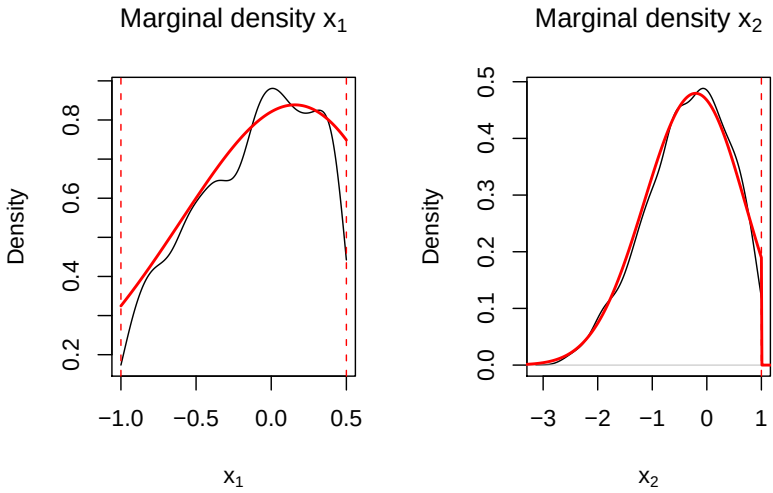


Figure 2. Marginal densities $F_k(x)$ $(k = 1, 2)$ for x_1 and x_2 obtained from Kernel density estimation of random samples and from direct evaluation of $F_k(x)$

which leads to the results $\boldsymbol{\mu}^* = (-0.152, -0.388)'$ and covariance matrix

$$\boldsymbol{\Sigma}^* = \begin{pmatrix} 0.163 & 0.161 \\ 0.161 & 0.606 \end{pmatrix}.$$

The trace plots in figures 3 and 4 show the evolution of a Monte Carlo estimate for the elements of the mean vector and the covariance matrix respectively for growing sample sizes. Furthermore, the 95% confidence interval obtained from Monte Carlo using the full sample of 10000 items is shown. All confidence intervals contain the true theoretical value, but Monte Carlo estimates still show substantial variation even with a sample size of 10000. Simulation from a truncated multivariate distribution and calculating the sample mean or the sample covariance also leads to consistent estimates of $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$. Since the rate of convergence of the MC estimator is $O(\sqrt{n})$, one has to ensure sufficient Monte Carlo iterations in order to have a good approximation or to choose variance reduction techniques.

Example 2

Let $\boldsymbol{\mu} = (0, 0, 0)'$,the covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.1 & 1.2 & 0 \\ 1.2 & 2 & -0.8 \\ 0 & -0.8 & 3 \end{pmatrix}$$

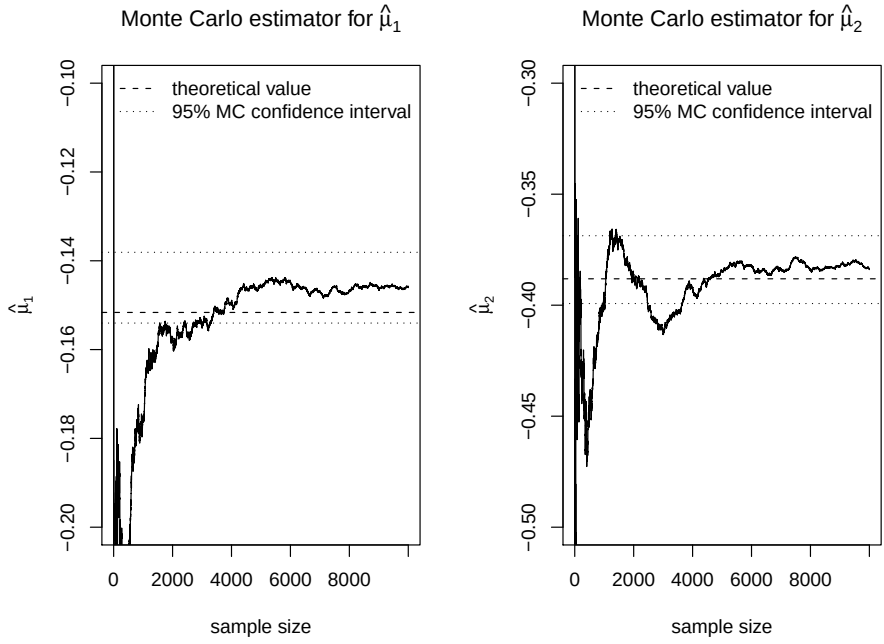


Figure 3. Trace plots of the Monte Carlo estimator for $\boldsymbol{\mu}^*$

and the lower and upper truncation points $\boldsymbol{a} = (-1, -\infty, -\infty)'$ and $\boldsymbol{b} = (0.5, \infty, \infty)'$. Then the only truncated variable is x_1, which is uncorrelated with x_3. Our formula results in $\boldsymbol{\mu}^* = c(-0.210, -0.229, 0)'$ and

$$\boldsymbol{\Sigma}^* = \begin{pmatrix} 0.174 & 0.190 & 0.0 \\ 0.190 & 0.898 & -0.8 \\ 0 & -0.8 & 3.0 \end{pmatrix}$$

For the special case of only $k < d$ truncated variables $(x_1, \ldots, x_k)$, the remaining $d - k$ variables $(x_{k+1}, \ldots, x_d)$ can be regressed on the truncated variables, and a simple formula for the mean and covariance matrix can be given (see Johnson & Kotz, 1971, p. 70).

Let the covariance matrix $\boldsymbol{\Sigma}$ of $(x_1, \ldots, x_d)$ be partitioned as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{pmatrix} \tag{24}$$

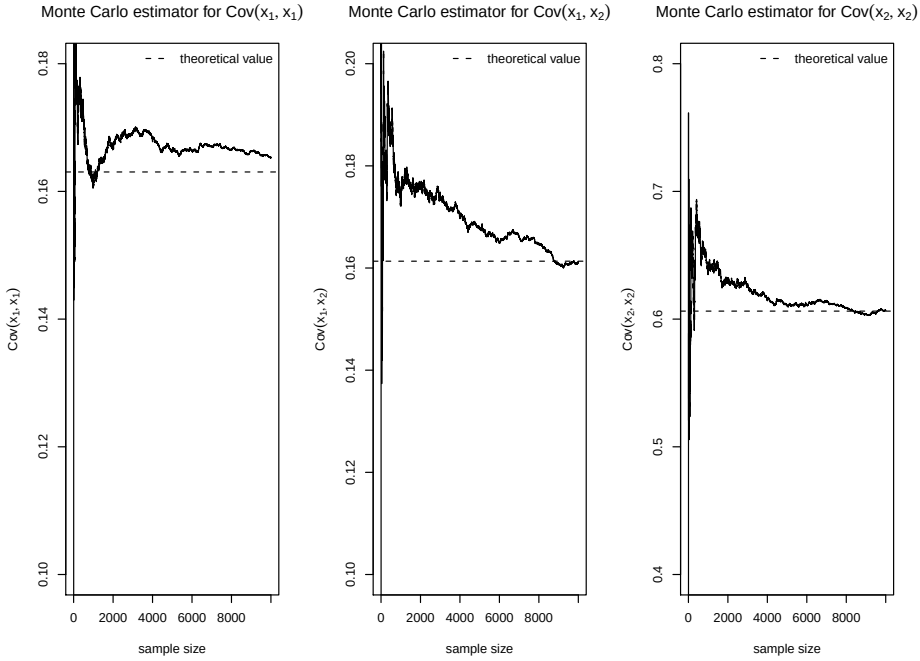


Figure 4. Trace plots of the Monte Carlo estimator for the 3 elements of $\boldsymbol{\Sigma}^*$ (σ^*_{11}, $\sigma^*_{12} = \sigma^*_{21}$ and σ^*_{22})

where $\mathbf{V}_{11}$ denotes the $k \times k$ covariance matrix of $(x_1, \ldots, x_k)$. The mean vector[3] and the covariance matrix $\boldsymbol{\Sigma}^*$ of all d variables can be computed as

$$(\boldsymbol{\xi}_1', \boldsymbol{\xi}_1' \mathbf{V}_{11}^{-1} \mathbf{V}_{12}) \tag{25}$$

and

$$\boldsymbol{\Sigma}^* = \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{11}\mathbf{V}_{11}^{-1}\mathbf{V}_{12} \\ \mathbf{V}_{21}\mathbf{V}_{11}^{-1}\mathbf{U}_{11} & \mathbf{V}_{22} - \mathbf{V}_{21}(\mathbf{V}_{11}^{-1} - \mathbf{V}_{11}^{-1}\mathbf{U}_{11}\mathbf{V}_{11}^{-1})\mathbf{V}_{12} \end{pmatrix} \tag{26}$$

where $\boldsymbol{\xi}_1'$ and $\mathbf{U}_{11}$ are the mean and covariance of the $(x_1, \ldots, x_k)$ after truncation.

The mean and standard deviation for the univariate truncated normal x_1 are

$$\xi_1 = \mu_1^* = \sigma_{11} \frac{\varphi_{\mu_1,\sigma_{11}}(a_1) - \varphi_{\mu_1,\sigma_{11}}(b_1)}{\Phi_{\mu_1,\sigma_{11}}(b_1) - \Phi_{\mu_1,\sigma_{11}}(a_1)},$$

$$\sigma_{11}^* = \sigma_{11} + \sigma_{11} \frac{a_1 \varphi_{\mu_1,\sigma_{11}}(a_1) - b_1 \varphi_{\mu_1,\sigma_{11}}(b_1)}{\Phi_{\mu_1,\sigma_{11}}(b_1) - \Phi_{\mu_1,\sigma_{11}}(a_1)}.$$

[3] The formula for the truncated mean given in Johnson and Kotz (1971, p. 70) is only valid for a zero-mean vector or after demeaning all variables appropriately. For non-zero means $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)'$ it will be $(\boldsymbol{\xi}_1', \boldsymbol{\mu}_2 + (\boldsymbol{\xi}_1' - \boldsymbol{\mu}_1)\mathbf{V}_{11}^{-1}\mathbf{V}_{12})$.

Letting $\mathbf{U}_{11} = \sigma^*_{11}$ and inserting ξ_1 and $\mathbf{U}_{11}$ into equations 25 and 26, one can verify our formula and the results for $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$. However, the crux in using the Johnson/Kotz formula is the need to first compute the moments of the truncated variables $(x_1, \ldots, x_k)$ for $k \geq 2$. But this has been exactly the subject of our paper.

6 Moment Calculation and Conditional Independence

In this section we establish a link between our moment calculation and the theory of graphical models (Edwards, 1995; Lauritzen, 1996; Whittaker, 1990). We present some properties of the inverse covariance matrix and show how the dependence structure of variables is affected after selection.

Graphical modeling uses graphical representations of variables as nodes in a graph and dependencies among them as edges. A key concept in graphical modeling is the conditional independence property. Two variables x and y are conditionally independent given a variable or a set of variables z (notation $x \perp\!\!\!\perp y|z$), when x and y are independent after partialling out the effect of z. For conditionally independent x and y, the edge between them in the graph is omitted and the joint density factorizes as $f(x, y|z) = f(x|z)f(y|z)$.

Conditional independence is equivalent to having zero elements $\boldsymbol{\Omega}_{xy}$ in the inverse covariance matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ as well as having a zero partial covariance/correlation between x and y given the remaining variables:

$$x \perp\!\!\!\perp y|\text{Rest} \iff \boldsymbol{\Omega}_{xy} = 0 \iff \rho_{xy.Rest} = 0.$$

Both marginal independence and conditional independence between variables simplify the computations of the truncated covariance in equation 16. In the presence of conditional independence of i and j given q, the terms $\sigma_{ij} - \sigma_{iq}\sigma_{qq}^{-1}\sigma_{qj} = 0$ vanish as they reflect the partial covariance of i and j given q.

As has been shown by Marchetti and Stanghellini (2008), the conditional independence property is preserved after selection, i.e. the inverse covariance matrices $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}^*$ before and after truncation share the same zero-elements. We prove that many elements of the precision matrix are invariant to truncation. For the case of $k < d$ truncated variables, we define the set of truncated variables with $T = \{x_1, \ldots, x_k\}$, and the remaining $d-k$ variables as $S = \{x_{k+1}, \ldots, x_d\}$. We can show that the off-diagonal elements $\boldsymbol{\Omega}_{i,j}$ are invariant after truncation for $i \in T \cup S$ and $j \in S$:

Proposition 1. *The off-diagonal elements $\boldsymbol{\Omega}_{i,j}$ and the diagonal elements $\boldsymbol{\Omega}_{j,j}$ are invariant after truncation for $i \in T \cup S$ and $j \in S$.*

Proof. The proof is a direct application of the Johnson/Kotz formula in equation 26 in the previous section. As a result of the formula for partitioned inverse

matrices (Greene, 2003, section A.5.3) , the corresponding inverse covariance matrix $\mathbf{\Omega}$ of the partitioned covariance matrix $\mathbf{\Sigma}$ is

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{V}_{11}^{-1}(\mathbf{I} + \mathbf{V}_{12}\mathbf{F}_2\mathbf{V}_{21}\mathbf{V}_{11}^{-1}) & -\mathbf{V}_{11}^{-1}\mathbf{V}_{12}\mathbf{F}_2 \\ -\mathbf{F}_2\mathbf{V}_{21}\mathbf{V}_{11}^{-1} & \mathbf{F}_2 \end{pmatrix} \tag{27}$$

with $\mathbf{F}_2 = (\mathbf{V}_{22} - \mathbf{V}_{21}\mathbf{V}_{11}^{-1}\mathbf{V}_{12})^{-1}$.

Inverting the truncated covariance matrix $\mathbf{\Sigma}^*$ in equation 26 using the formula for the partitioned inverse leads to the truncated precision matrix

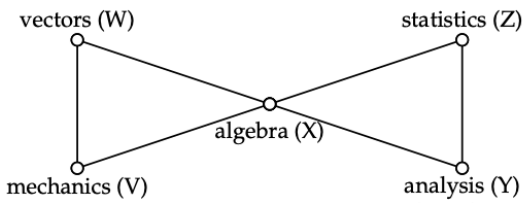
$$\mathbf{\Omega}^* = \begin{pmatrix} \mathbf{U}_{11}^{-1} + \mathbf{V}_{11}^{-1}\mathbf{V}_{12}\mathbf{F}_2\mathbf{V}_{21}\mathbf{V}_{11}^{-1} & -\mathbf{V}_{11}^{-1}\mathbf{V}_{12}\mathbf{F}_2 \\ -\mathbf{F}_2\mathbf{V}_{21}\mathbf{V}_{11}^{-1} & \mathbf{F}_2 \end{pmatrix} \tag{28}$$

where the $\mathbf{\Omega}_{12}^*$ and $\mathbf{\Omega}_{21}^*$ elements are the same as $\mathbf{\Omega}_{12}$ and $\mathbf{\Omega}_{21}$ respectively. The same is true for the elements in $\mathbf{\Omega}_{22}^*$, especially the diagonal elements in $\mathbf{\Omega}_{22}^*$. □

Here, we prove this invariance property only for a subset of truncated variables. Based on our experiments we conjecture that the same is true also for the case of full truncation (i.e. all off-diagonal elements in $\mathbf{\Omega}_{11}^*$). However, we do not give a rigorous proof here and leave it to future research.

Example 3

We illustrate the invariance of the elements of the inverse covariance matrix with the famous mathematics marks example used in Whittaker (1990) and Edwards (1995, p. 49). The independence graph of the five variables (W, V, X, Y, Z) in this example takes the form of a butterfly as shown in below.



Here, we have the conditional independence $(W, V) \perp\!\!\!\perp (Y, Z)|X$. A corresponding precision matrix might look like (sample data; zero-elements marked as "."):

$$\mathbf{\Omega} = \begin{pmatrix} 1 & 0.2 & 0.3 & . & . \\ 0.2 & 1 & -0.1 & . & . \\ 0.3 & -0.1 & 1 & 0.4 & 0.5 \\ . & . & 0.4 & 1 & 0.2 \\ . & . & 0.5 & 0.2 & 1 \end{pmatrix} \tag{29}$$

After truncation in some variables (for example (W, V, X) as $-2 \leq W \leq 1$, $-1 \leq V \leq 1$, $0 \leq X \leq 1$), we apply equation 16 to compute the truncated second moment and the inverse covariance matrix as:

$$\boldsymbol{\Omega}^* = \begin{pmatrix} 1.88 & 0.2 & 0.3 & . & . \\ 0.2 & 3.45 & -0.1 & . & . \\ 0.3 & -0.1 & 12.67 & 0.4 & 0.5 \\ . & . & 0.4 & 1 & 0.2 \\ . & . & 0.5 & 0.2 & 1 \end{pmatrix} \tag{30}$$

The precision matrix $\boldsymbol{\Omega}^*$ after selection differs from $\boldsymbol{\Omega}$ only in the diagonal elements of (W, V, X). From $\boldsymbol{\Omega}^*$, we can read how partial correlations between variables have changed due to the selection process.

Each diagonal element $\boldsymbol{\Omega}_{yy}$ of the precision matrix is the inverse of the partial variance after regressing on all other variables (Whittaker, 1990, p. 143). Since only those diagonal elements in the precision matrix for the $k \leq d$ of the truncated variables will change after selection, this leads to the idea to just compute these k elements after selection rather than the full $k(k+1)/2$ symmetric elements in the truncated covariance matrix and applying the Johnson/Kotz formula for the remaining $d - k$ variables. However, the inverse partial variance of a scalar y given the remaining variables $X = \{x_1, \ldots, x_d\} \setminus y$

$$\boldsymbol{\Omega}^*_{yy} = \left[\Sigma^*_{y.X}\right]^{-1} = \left[\Sigma^*_{yy} - \Sigma^*_{yX}\Sigma^{*-1}_{XX}\Sigma^*_{Xy}\right]^{-1}$$

still requires the truncated covariance results derived in Section 3.

7 Summary

In this paper, we derived a formula for the first and second moments of the doubly truncated multivariate normal distribution and for their bivariate marginal density. An implementation for both formulas has been made available in the R statistics software as part of the `tmvtnorm` package. We linked our results to the theory of graphical models and proved an invariance property for elements of the precision matrix. Further research can deal with other types of truncation (e.g. elliptical). Another line of research can look at the moments of the doubly truncated multivariate Student-t distribution, which contains the truncated multivariate normal distribution as a special case.

References

Amemiya, T. (1974). Multivariate regression and simultaneous equations models when the dependent variables are truncated normal. *Econometrica*, *42*, 999–1012. doi: https://doi.org/10.2307/1914214

Arellano-Valle, R. B., & Genton, M. G. (2005). On fundamental skew distributions. *Journal of Multivariate Analysis*, *96*, 93–116. doi: https://doi.org/10.1016/j.jmva.2004.10.002

Arismendi, J. C. (2013). Multivariate truncated moments. *Journal of Multivariate Analysis*, *117*, 41–75. doi: https://doi.org/10.1016/j.jmva.2013.01.007

Cartinhour, J. (1990). One-dimensional marginal density functions of a truncated multivariate normal density function. *Communications in Statistics - Theory and Methods*, *19*, 197–203. doi: https://doi.org/10.1080/03610929008830197

Edwards, D. (1995). *Introduction to graphical modelling.* Springer.

Galarza C.E., R., Kan, & V.H., L. (2021). *MomTrunc: Moments of folded and doubly truncated multivariate distributions.* Retrieved from `https://cran.r-project.org/web/packages/MomTrunc` (R package version v5.97)

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, *1*, 141–149. doi: https://doi.org/10.2307/1390838

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., & Hothorn, T. (2012). *mvtnorm: Multivariate normal and t distributions.* Retrieved from `http://CRAN.R-project.org/package=mvtnorm` (R package version 0.9-9992)

Greene, W. H. (2003). *Econometric analysis* (5th ed.). Prentice-Hall.

Gupta, A. K., & Tracy, D. S. (1976). Recurrence relations for the moments of truncated multinormal distribution. *Communications in Statistics - Theory and Methods*, *5*(9), 855–865. doi: https://doi.org/10.1080/03610927608827402

Johnson, N. L., & Kotz, S. (1971). *Distributions in statistics: Continuous multivariate distributions.* John Wiley & Sons.

Kan, R., & Robotti, C. (2017). On moments of folded and truncated multivariate normal distributions. *Journal of Computational and Graphical Statistics*, *26(4)*, 930–934. doi: https://doi.org/10.1080/10618600.2017.1322092

Lauritzen, S. (1996). *Graphical models.* Oxford University Press.

Lee, L.-F. (1979). On the first and second moments of the truncated multinormal distribution and a simple estimator. *Economics Letters*, *3*, 165–169. doi: https://doi.org/10.1016/0165-1765(79)90111-3

Lee, L.-F. (1983). The determination of moments of the doubly truncated multivariate tobit model. *Economics Letters*, *11*, 245–250.

Leppard, P., & Tallis, G. M. (1989). Algorithm AS 249: Evaluation of the mean and covariance of the truncated multinormal distribution. *Applied Statistics*, *38*, 543–553. doi: https://doi.org/10.2307/2347752

Marchetti, G. M., & Stanghellini, E. (2008). A note on distortions induced by truncation with applications to linear regression systems. *Statistics & Probability Letters*, *78*, 824–829. doi: https://doi.org/10.1016/j.spl.2007.09.050

Muthén, B. (1990). Moments of the censored and truncated bivariate normal distribution. *British Journal of Mathematical and Statistical Psychology*,

43, 131–143. doi: https://doi.org/10.1111/j.2044-8317.1990.tb00930.x

Regier, M. H., & Hamdan, M. A. (1971). Correlation in a bivariate normal distribution with truncation in both variables. *Australian Journal of Statistics*, *13*, 77–82. doi: https://doi.org/10.1111/j.1467-842x.1971.tb01245.x

Rosenbaum, S. (1961). Moments of a truncated bivariate normal distribution. *Journal of the Royal Statistical Society. Series B (Methodological)*, *23*, 405-408. doi: https://doi.org/10.1111/j.2517-6161.1961.tb00422.x

Shah, S. M., & Parikh, N. T. (1964). Moments of single and doubly truncated standard bivariate normal distribution. *Vidya (Gujarat University)*, *7*, 82–91.

Tallis, G. M. (1961). The moment generating function of the truncated multinormal distribution. *Journal of the Royal Statistical Society, Series B (Methodological)*, *23*(1), 223–229. doi: https://doi.org/10.1111/j.2517-6161.1961.tb00408.x

Tallis, G. M. (1963). Elliptical and radial truncation in normal populations. *The Annals of Mathematical Statistics*, *34*(3), 940–944. doi: https://doi.org/10.1214/aoms/1177704016

Tallis, G. M. (1965). Plane truncation in normal populations. *Journal of the Royal Statistical Society, Series B (Methodological)*, *27*(2), 301–307. doi: https://doi.org/10.1111/j.2517-6161.1965.tb01497.x

Whittaker, J. (1990). *Graphical models in applied multivariate statistics.* John Wiley & Sons.

Wilhelm, S., & Manjunath, B. G. (2010, June). tmvtnorm: A Package for the Truncated Multivariate Normal Distribution. *The R Journal*, *2*(1), 25–29. doi: https://doi.org/10.32614/rj-2010-005

Wilhelm, S., & Manjunath, B. G. (2012). *tmvtnorm: Truncated multivariate normal distribution and Student t distribution.* Retrieved from `http://CRAN.R-project.org/package=tmvtnorm` (R package version 1.4-5)

Journal of Behavioral Data Science, 2021, *1* (1), 34–52.
DOI: https://doi.org/10.35566/jbds/v1n1/p3

Birds of a Feather Flock Together and Opposites Attract: The Nonlinear Relationship Between Personality and Friendship

Haiyan Liu[1] and Zhiyong Zhang[2]

[1] University of California-Merced, Merced, CA 95343, USA
hliu62@ucmerced.edu
[2] University of Notre Dame, Notre Dame, IN 46556, USA
zzhang4@nd.edu

Abstract. Whether birds of a feather flock together or opposites attract is a classical research question in social and personality psychology. In most existing studies, correlation-based techniques are commonly used to study the similarity/dissimilarity among social entities. Social network data comprises two primary components: actors and the possible social relations between them. It, therefore, has observations on both the dyads with and without social relations. Because of the availability of the baseline group (dyads without social relations), it is possible to contrast the two groups of dyads using social network analysis techniques. This study aims to illustrate how to use social network analysis techniques to address psychological research questions. Specifically, we will investigate how the similarity or dissimilarity of actor's characteristics relates to the likelihood for them to build social relations. By analyzing a college friendship network, we found the quadratic relations between personality similarity and friendship. Both very similar and very dissimilar personalities boost friendship among college students.

Keywords: Friendship network · Personality · Social network analysis · Quadratic relation · Factor analysis

1 Introduction

Social relations play a crucial role in an individual's social and behavioral development (Cacioppo & Cacioppo, 2014; House, Landis, & Umberson, 1988; McCamish-Svensson, Samuelsson, Hagberg, Svensson, & Dehlin, 1999; Umberson, Crosnoe, & Reczek, 2010). Close and healthy social relations benefit people's subjective well-being in their life span (McCamish-Svensson et al., 1999; Seeman, 2001; Waldinger, Cohen, Schulz, & Crowell, 2015). Social relations also impact people's health behavior such as alcohol use (Balsa, Homer, French, &

Norton, 2011). Understanding and predicting the formation of social relations is thus of enormous interests to researchers and has been traditionally studied in social and personality psychology (e.g., Bahns, Crandall, Gillath, & Preacher, 2017; Cacioppo & Cacioppo, 2014)

In the existing literature, the principle of homophily is "believed" to be the mainstream of the formation of social relations. In other words, individuals in close social relations share many similar characteristics (McPherson, Smith-Lovin, & Cook, 2001; Rushton & Bons, 2005). A large body of research has investigated the presence of similar personality attributes in close relations such as romantic relations and friendships (e.g., Asendorpf & Wilpers, 1998; Harris & Vazire, 2016; Liu, Jin, & Zhang, 2018; Youyou, Stillwell, Schwartz, & Kosinski, 2017). Much of the research found no or weak personality similarity (Altmann, Sierau, & Roth, 2013; Watson, Beer, & McDade-Montez, 2014; Watson, Hubbard, & Wiese, 2000). Others found moderate similarities in some of the Big Five personality factors (McCrae et al., 2008). Youyou et al. (2017) revealed personality similarity among couples and friends. Another study found that individuals tended to select those with similar personalities as friends (Bahns et al., 2017). Hudson and Fraley (2014) found a quadratic relationship between partners' personality-trait-similarity and relationship satisfaction among people with low avoidance and high anxiety. The existing conclusions seem to be inconclusive.

There are at least two potential reasons that account for the inconsistency in the literature. In most of these studies, only data on dyads are available due to the data collection methods such as collecting data from friends whereas data on dyads without social relations are not available. Therefore, few of these studies actually contrasted the two types of dyads due to the lack of the baseline group. Moreover, correlation analysis is the dominant approach used in studying the similarities of two actors forming dyads, which only focuses on the linear relationship between two variables and oversights the potential nonlinear relationships.

Social network data, however, contain both dyads with social relations and dyads without social ties. A social network comprises a group of actors and the potential relationship between them (Wasserman & Faust, 1994). In a network graph $\mathbf{M}$, nodes represent "actors," and they could be any entities such as students in a friendship network, research institutions in a collaboration network, and variables in a variable network (Epskamp, Rhemtulla, & Borsboom, 2017). The ties/edges in a network display the relations, interactions or dependence among "actors." It thus provides a premise to study the association between actor attribute similarity and social relations as in previous studies. It further allows researchers to compare two types of dyads using tools other than correlation analysis. It potentially leads to more interpretable results. In recent years, efforts have been made to address social and psychological research questions from the network perspective. Sweet (2016) reviewed common descriptive methods and network models for educational and psychological research. Clifton and Webster (2017) discussed the use of social network data

to address psychological research questions through several examples. Liu et al. (2018) proposed a structural equation model to predict the existence of binary social relations using the latent personality distance.

The goal of the present work is multifold. First, we introduce some measures to quantify dyads' properties, which are named "nodal/dyadic" covariates. These measures are not necessarily about similarity but could be in any meaningful format. Second, we demonstrate how to use the newly introduced measures to predict social relations using the proposed model by Liu et al. (2018), which provides a primer on predicting social bonds in a network. Third, we illustrate how to conduct the model selection and choose the model that fits the data best.

The rest of this article is structured as follows. First, we describe the college friendship network data collected by the Lab of Big Data at the University of Notre Dame. Next, we explore the factor structures of personality data. We then predict a valued friendship network using student's characteristics and select the model that fits the data best. In the end, we conclude the study with discussions on the current development and future directions.

2 Friendship Network: An Empirical Example

Throughout this paper, we use the data collected by the Lab of Big Data at the University of Notre Dame (Liu et al., 2018).

2.1 Participants

The participants are 162 students in a 4-year college in China. All the students were studying at the school of art and letters while completing the survey. Therefore, the boundary of the friendship network was known before data collection. Among the 162 students, there were 90 female and 72 male students. Their average age was 21.64 years (SD=0.86).

2.2 Procedures and Measures

Four types of information are available: (1) friendship networks, (2) actor attributes including demographic information, (3) behaviors, and (4) personalities.

2.2.1 Friendship networks To collect the network data, we gave each student a roster of all the 162 students and asked them to report their acquaintanceship with every other student. The friendship was measured on a 5-point *Likert* scale ranging from "I have never heard about this student." to "The person is one of my best friends." (See Table 1). In the current study, we used the maximal relationship between a pair of students. If two students have different evaluations on the friendship between them, we use the stronger evaluation. Therefore, the relationship is symmetric and non-directional. With

Table 1. 5-point Likert scale for the friendship

Level	Meaning
0	I have never heard the name.
1	I heard about the person but had no personal interaction with her/him.
2	I have met the person a few times but he/she is not a friend of mine.
3	The person is a friend of mine.
4	The person is one of my best friends.

162 students, the network data are recorded in a 162 by 162 matrix **M**, which is called a "sociomatrix" in the field of social network analysis. A row of **M** contains the responses of the row actor on their friendship relations with the column actors.

A plot of the friendship network with ordinal relations is included in Figure 1. In the heatmap of the friendship network, a darker square represents a stronger relationship between the students in the corresponding row and column. On the diagonal from the bottom left to the upright, there are six blocks standing out with dark color, each containing a group of students with closer relations. Those blocks are clusters of the college student friendship network.

2.2.2 Personality We used the 20-item Mini-IPIP Scale for the Big Five factors of personality (Donnellan, Oswald, Baird, & Lucas, 2006). The five factors measured include Intellect/Imagination (or Openness), Conscientiousness, Extraversion, Agreeableness, and Neuroticism. Each of the five factors is measured by 4 items. Example items of the Mini-IPIP scale are: "In general, I am the life of the party" and "I am not interested in abstract ideas." The 20 items were rated on a 5-point Likert scale (i.e., 1 = strongly disagree, 2 = somewhat disagree, 3 = neither agree nor disagree, 4 = somewhat agree, and 5 = strongly agree). For reverse coded items, the scores were reversed before analysis.

2.2.3 Actor Attribute Data Participants also reported data on their behaviors. Participants rated themselves on these items using a true or false format. To collect data on the alcohol use, each student reported whether they had drunk alcohol in the past 30 days or not. Among the 162 students, 68 students reported they have drunk alcohol in the past thirty days. Besides, information on academic performance was also available, with scores ranging from 18 to 87. The average academic performance score was 54.99, with a standard deviation of 10.94.

2.3 Overview of Data Analysis

The purpose of the analysis is to exemplify the potentials of social network analysis in psychological research. Specifically, we will investigate how personality predicts friendship. In the literature, there are arguments on both

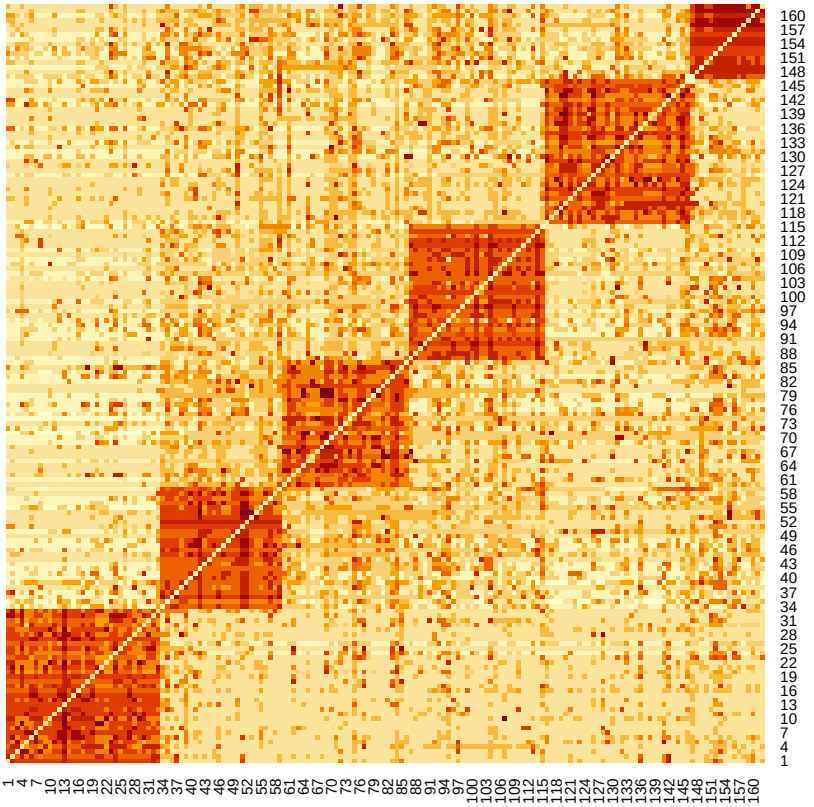


Figure 1. Heatmap of the friendship network. Darker color indicates a higher level friendship

"Birds of a feather flock together," and "Opposites attract." If birds of a feather flock together, then we can expect that students with similar personality traits should be more likely to be friends. If opposites attract, then we can expect those with dissimilar personalities should boost the likelihood for them to be friends. If both statements are plausible, then we should expect a nonlinear relation between personality similarity and friendship. In the following, we will first explore the factor structures of personalities.

3 Factor Extraction

We conducted a confirmatory factor analysis (CFA, Cattell, 1952) to evaluate the structure of the latent personality traits. The reliability (α) of the five scales are 0.57 for "intelligence/imagination", 0.48 for "conscientiousness", 0.62 for "extraversion", 0.48 for "agreeableness", and 0.40 for "neuroticism." We decided to use two factors−imagination and extraversion−in the CFA because they have relatively high α values. Let $\boldsymbol{\eta}$ be the vector of latent personality factors and $\mathbf{w}$ be their indicators. The CFA model has the following general form,

$$\begin{cases} \mathbf{w}_i & = \boldsymbol{\Lambda}\boldsymbol{\eta}_i + \boldsymbol{\varepsilon}_i \\ \boldsymbol{\eta}_i & \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Phi}) \\ \boldsymbol{\varepsilon}_i & \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Psi}), \end{cases} \tag{1}$$

where $\mathbf{w}_i$ is the indicator data on actor i, $\boldsymbol{\varepsilon}_i$ is a $J \times 1$ vector of unique factors and it follows a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Psi}$. The factor loading matrix $\boldsymbol{\Lambda}$ is a $J \times D$ matrix. $\boldsymbol{\Phi}$ is the factor covariance matrix to be estimated. In this model, the unknowns include individuals' factor scores $\{\boldsymbol{\eta}_i\}_{i=1}^N$ and model parameters $\{\boldsymbol{\Lambda}, \boldsymbol{\Phi}, \boldsymbol{\Psi}\}$. We fix one factor loading of each factor to be 1 for the purpose of model identification.

We conducted model modification after fitting the model without cross-loadings and correlations among items to explore the factor structure. We ended up with the final model with RMSEA 0.047 and CFI 0.963. The path diagram of the final model is presented in Figure 2.

Recall that the purpose of the current study was to investigate the association between personality similarity and friendships. We, therefore, recorded estimates for both the factor covariance matrix $\boldsymbol{\Phi}$ and individuals' factor scores $\boldsymbol{\eta}_i$, which will be used to compute the personality similarity (i.e., distance) of any two students. The estimated factor covariance matrix is provided in Table 2.The variance estimates of extraversion and imagination are 0.838 and 0.252, respectively, and their covariance is 0.172.

Table 2. Estimated variance and covariance of latent factors

cov(,)	Extraversion	Imagination
Extraversion	0.838	0.172
Imagination	0.172	0.252

Despite many factor score estimators, the Thurstone-Thomson "regression" factor scores (Thurstone, 1935) were extracted and used in the subsequent analysis following the recommendations by both Devlieger, Mayer, and Rosseel (2016) and Liu et al. (2018). The scatterplot and the histograms of the predicted factor scores are provided in Figure 3. Each dot in Figure 3 represents the location of a student in the personality space formed by the scores of extraversion and imagination. Two students sharing similar personality traits in extraversion and imagination would stay close to each other in the personality space.

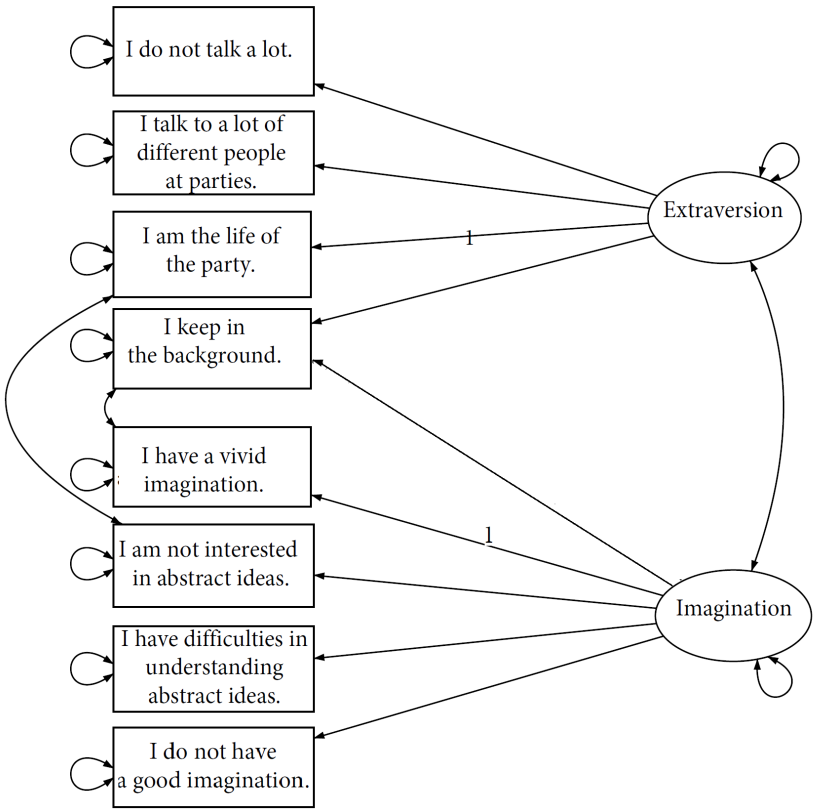


Figure 2. Path diagram of the CFA model of Imagination and Extraversion

4 Probit Model for Ordinal Networks

The model we will introduce is built on the prior work on structural equation modeling of social networks by Liu et al. (2018). In this modeling framework, individuals are assumed to hold a position in a latent space formed by personality traits (i.e., personality space). The distance/(dis)similarity between two individuals in the personality space predicts how likely they connect in the manifest social world. This modeling framework is developed to predict social relations using individuals' characteristics. This model can particularly investigate whether similar personalities or dissimilar personalities boost friendships among college students.

In the following, we will present the model in a form for analyzing networks with ordinal relations and demonstrate its applications in examining the relationship between personality similarity and friendships. We will compare the following plausible hypotheses: (1) similar personality traits promote friendship;

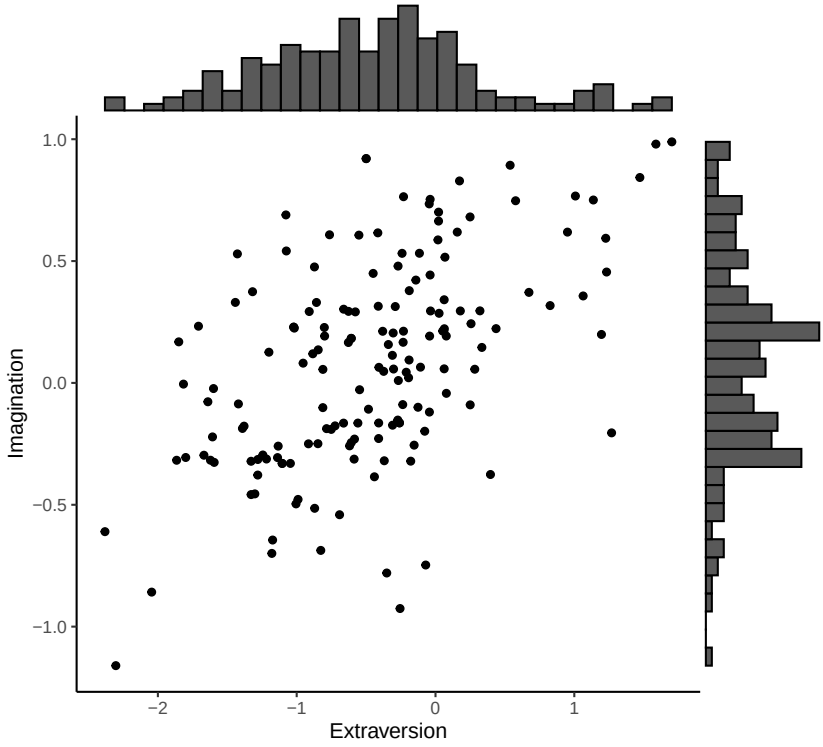


Figure 3. Predicted factor score

(2) dissimilar personality traits imply a higher chance to be friends; or (3) both are plausible.

The data analysis will use a three-phase procedure. First, we will define "nodal covariates" (i.e., dyads level covariates) based on the research hypotheses of interests. Second, we will build a Probit model to investigate how the nodal covariates predict friendship. Third, we will conduct likelihood ratio tests to select the model with the best fit for the data.

4.1 Nodal Covariates

The study focuses on predicting the ordinal ties in the friendship network, which is a dyadic level analysis of social networks. Therefore, we need to construct dyadic covariates describing the characteristics of a pair of students. In addition to personality traits, we also consider three manifest covariates-gender, academic performance, and class membership.

Same-gender friendship has been of interest to researchers (Benenson, 1990; Elkins & Peterson, 1993; Jones, 1991; Zarbatany, Conley, & Pepper, 2004). To

test the effect of gender on friendship, we define the following nodal covariate,

$$h_{\text{gender}}(i,j) = \begin{cases} 1 & \text{if students } i \text{ and } j\text{are of the same gender} \\ 0 & \text{otherwise.} \end{cases}$$

Using the nodal covariate h_{gender}, we can study the homogeneous gender effect on the acquaintance levels.

Academic achievement is measured using a continuous scale. To quantify the similarity in academic achievement, we define a nodal covariate of academic achievement as the absolute difference of two students' scores

$$h_{\text{score}}(i,j) = |\text{score}_i - \text{score}_j|.$$

The larger value on $h_{\text{score}}(i,j)$, the more discrepancy of students i and j on their academic achievement.

The 162 students participating in our study belonged to different "classes." Students from the same class take the same courses more often, and potentially have more chances to build friendships. Therefore, we control the class membership effect in our analysis. The nodal covariate of class membership takes value one if two students are from the same class and 0 otherwise. That is

$$h_{\text{class}}(i,j) = \begin{cases} 1 & \text{if students } i \text{ and } j \text{ are from the same class,} \\ 0 & \text{otherwise.} \end{cases}$$

In addition to the three manifest nodal covariates h_{gender}, h_{score}, and h_{class}, we focus on the relationship between the personality similarity and friendships. To quantify the personality similarity, we use the Mahalanobis distance (Mahalanobis, 1936) of the personality factor scores of two students,

$$d_{ij} = h_{\text{personality}}(i,j) = \sqrt{(\boldsymbol{\eta}_i - \boldsymbol{\eta}_j)^t \boldsymbol{\Phi}^{-1} (\boldsymbol{\eta}_i - \boldsymbol{\eta}_j)}, \tag{2}$$

where $\boldsymbol{\eta}_i$ and $\boldsymbol{\eta}_j$ are the vectors of personality factor scores of students i and j, and $\boldsymbol{\Phi}$ is the covariance matrix of personality latent factors. The Mahalanobis distance is the standardized distance of two correlated vectors penalized by the covariance between them.

We want to note that the concept of the "nodal" covariate is flexible to include any statistics that summarize the information of dyads. Researchers can define their nodal covariates based on their research hypothesis. Moreover, a nodal covariate is not necessarily capturing the similarity of actors as exemplified. Instead, it could be of any type. To provide an example, one can define overall academic achievement as the sum of scores of two students and test whether the overall score relates to the friendship or not. Instead of studying the effect of similar personality, one could also study the overall extraversion level of two students and investigate its impact on the friendship between the two students.

4.2 Probit Regression Analysis of Ordinal Networks

To model the association between personality similarity and friendship, we extended the work by Liu et al. (2018) to undirected valued networks with ordinal relations. A probit model is adopted to predict the ordinal relations using nodal covariates (Agresti, 2013). Let m_{ij} be the level of friendship between student i and j. It could take one of the five ordinal values 0, $1, 2, 3$, or 4 in the college friendship introduced in the previous section. A greater value indicates a stronger relationship between the two students. For a level $k = 0, 1, 2, \cdots, 4$, let $\pi_{ij}^{(k)}$ be the probability for m_{ij} to be in the k'th category,

$$p(m_{ij} = k) = \pi_{ij}^{(k)}, \qquad \text{for } k = 0, 1, \cdots, 4. \tag{3}$$

The cumulative probability for a tie in a category k and below is

$$p(m_{ij} \leq k) = \pi_{ij}^{(0)} + \pi_{ij}^{(1)} + \cdots + \pi_{ij}^{(k)}, \qquad \text{for } k = 0, 1, 2, \cdots, 4 \tag{4}$$

and $\sum_{k=0}^{4} \pi_{ij}^{(k)} = 1$, since any friendship tie must fall in one of the five categories. To predict the probability for a tie to fall in a category using nodal statistics on dyads, we use an ordered probit model,

$$\begin{cases} \text{Probit } \left[p(m_{ij} \leq k)\right] & = F^{-1}\left[p(m_{ij} \leq k)\right] \qquad \text{for } k = 0, 1, \cdots, 3 \\ & = \tau_{k|k+1} - (\boldsymbol{\beta}' \mathbf{h}_{ij} + \gamma d_{ij}) \\ \pi_{ij}^{(4)} & = 1 - \sum_{k=0}^{3} \pi_{ij}^{(k)} \end{cases} \tag{5}$$

where $F(\cdot)$ is the cumulative density function (CDF) of the standard normal distribution (i.e., N(0, 1)), and d is the latent personality distance computed as $d = \sqrt{(\boldsymbol{\eta}_i - \boldsymbol{\eta}_j)^t \boldsymbol{\Phi}^{-1} (\boldsymbol{\eta}_i - \boldsymbol{\eta}_j)})$ as in Equation (2). The parameters $\boldsymbol{\beta}$ and γ are coefficients of manifest nodal covariates and latent factor distance (i.e., d). Because $F^{-1}(\cdot)$ is an increasing function, the intercept coefficients must follow an ordered sequence,

$$\tau_{0|1} \leq \tau_{1|2} \leq \cdots \leq \tau_{3|4}.$$

To further understand the impact of the slope parameter γ on the propensities of categories, four plots with different values for γ are provided in Figure 4. We generate data from a model with four categories, and the three thresholds are $\tau_{0|1} = -1$, $\tau_{1|2} = 0$, and $\tau_{2|3} = 1$ and one manifest covariate (i.e., $h1$) whose coefficient $\beta = 0.6$. Given $h1 = 0$, we computed the implied cumulative probabilities with varying d. In Figure 4, the red, green, blue, and purple curves are the probability for a tie in category 0, category 0 or 1, category 0, 1, or 2, and category 0, 1, 2, or 3.

First, when $\gamma < 0$ (Plot (a) and (b) in Figure 4), the cumulative probabilities are increasing as the latent distance d increases. Thus, the probability for a tie in a higher-level category decreases. When $\gamma > 0$ (Plot (c) and (d)), the trajectories of the cumulative probability are in the opposite direction. A positive value of γ indicates that with a larger latent distance d, the probability for a relationship to be in a higher-level category increases. The magnitude of γ (i.e., $|\gamma|$) tells the

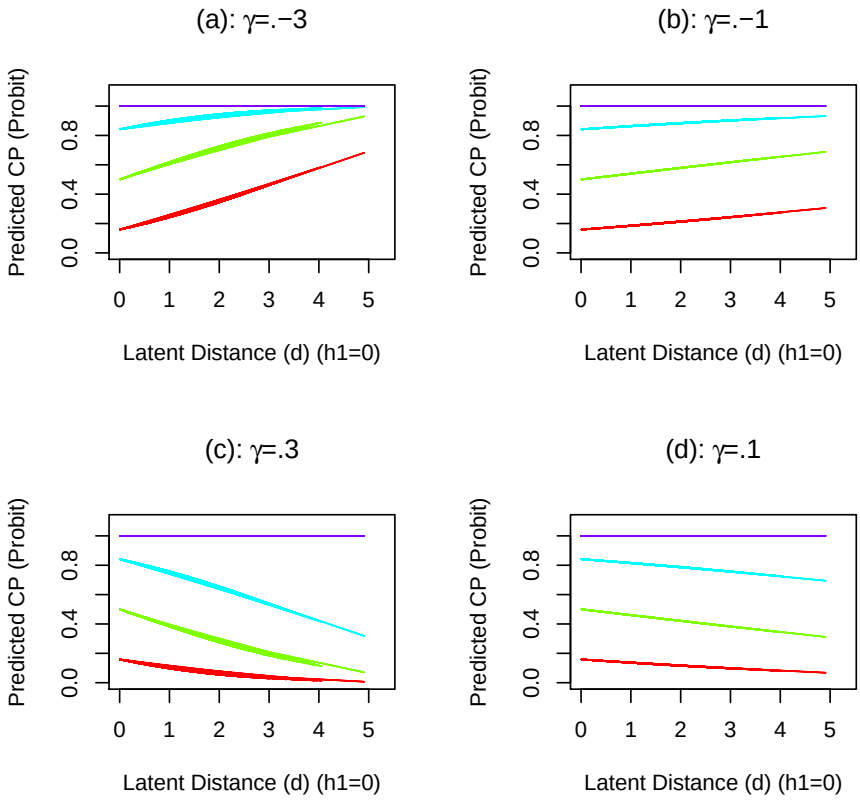


Figure 4. Plots of cumulative probabilities (CP) with different slope parameters in a model with 4 ordinal levels given the level of other covariates. The red, green, blue, and purple curves are the CP up to category 0, 1, 2, and 3.

extent to which the latent distance affects the cumulative probability. A larger $|\gamma|$ implies stronger impacts of latent distance d on the friendship.

To investigate the potential higher-order relationship between the personality similarity and friendship, we fit the model with the quadratic term of personality distance. To check whether the quadratic model is the conclusive model, we can fit the model with the cubic term of the personality distance. Therefore, we fit three competing models: a linear model with the first-order distance, i.e., d_{ij} as a predictor, a quadratic model with d_{ij}^2 as a predictor, and a cubic model with d_{ij}^3 as a predictor.

4.2.1 Linear Probit Model In the linear probit model, we include three manifest nodal covariates h_{gender}, h_{score}, and h_{class} as well as the latent

personality distance d,

$$\begin{cases} P(m_{ij} \le k) & = \pi_{ij}^0 + \pi_{ij}^1 + \cdots + \pi_{ij}^{(k)}, \qquad \text{for } k = 0, 1, \cdots, 4. \\ \text{Probit } \big[P(m_{ij} \le k)\big] & = F^{-1}\big[P(m_{ij} \le k)\big] \qquad \text{for } k = 0, 1, \cdots, 3 \\ & = \tau_{k|k+1} - (\beta_1 h_{gender}(i,j) + \beta_2 h_{score}(i,j) \\ & \quad + \beta_3 h_{class}(i,j) + \gamma d_{ij}) \\ \pi_{ij}^{(4)} & = 1 - \sum_{k=0}^{3} \pi_{ij}^{(k)} \end{cases} \tag{6}$$

In this model, the coefficient γ explains the extent to which the personality distance d_{ij} predicts friendship. With a negative γ, the probability of having a higher level of friendship is greater when d_{ij} is smaller, so the more similar personalities associate with a higher chance to have a closer friendship. If γ is positive, then dissimilar personalities boost friendship.

4.2.2 Quadratic Probit Model In the second model, we also include a quadratic term of the latent personality distance, and the model becomes,

$$\begin{aligned} \text{Probit } \big[P(m_{ij} \le k)\big] &= F^{-1}\big[P(m_{ij} \le k)\big] \qquad \text{for } k = 0, 1, \cdots, 3 \\ &= \tau_{k|k+1} - (\beta_1 h_{gender}(i,j) + \beta_2 h_{score}(i,j) + \beta_3 h_{class}(i,j) \\ &\quad + \gamma_1 d_{ij} + \gamma_2 d_{ij}^2). \end{aligned} \tag{7}$$

This model is useful for investigating the potential quadratic relationship between the personality similarity and friendship, and it also helps identify the transition points of the trend.

4.2.3 Cubic Probit Model The cubic model includes the third-order of the distance, i.e., d_{ij}^3, in the analysis,

$$\begin{aligned} \text{Probit } \big[P(m_{ij} \le k)\big] &= F^{-1}\big[P(m_{ij} \le k)\big] \qquad \text{for } k = 0, 1, \cdots, 3 \\ &= \tau_{k|k+1} - (\beta_1 h_{gender}(i,j) + \beta_2 h_{score}(i,j) + \beta_3 h_{class}(i,j) \\ &\quad + \gamma_1 d_{ij} + \gamma_2 d_{ij}^2 + \gamma_3 d_{ij}^3) \end{aligned} \tag{8}$$

By fitting the cubic model, we can investigate if there is more than one transition point for the relationship between personality similarity and friendship.

To estimate the model, we first evaluate the factor structure of the extroversion and imagination, and obtain the model parameter estimates and the Thurstone-Thomson "regression" factor scores $\hat{\boldsymbol{\eta}}_i$ and $\hat{\boldsymbol{\eta}}_j$ as discussed in the previous section. We then compute the estimated personality distance

$$\hat{d}_{ij} = \sqrt{(\hat{\boldsymbol{\eta}}_i - \hat{\boldsymbol{\eta}}_j)^t \hat{\boldsymbol{\Phi}}^{-1} (\hat{\boldsymbol{\eta}}_i - \hat{\boldsymbol{\eta}}_j)}.$$

According to the suggestions by Liu et al. (2018), the use of Thurstone-Thomson factor scores led to asymptotically unbiased estimates for the γ parameter.

5 Result

In this section, we will present the results of the three models discussed in the previous section

5.1 Model Selection

To evaluate the relative performance of the three models (i.e., linear, quadratic, and cubic probit model), we conducted likelihood ratio tests using the saved deviance in Table 3. For the linear model against the quadratic model, the Chi-square statistic is 9.514 and with a p-value of .002. Hence, the quadratic model is significantly better than the linear model. When the quadratic model is compared against the cubic (third-order) model, the Chi-square statistic is 0.318 with a p-value of .573. Thus, the cubic model is not significantly better than the quadratic model. The quadratic model is thus the best model.

Table 3. Likelihood ratio test of the three nested models

	Model	Deviance	Test	Df	LR Stat	Pr(Chi)
1	Linear	28560.55				
2	Quadratic	28551.03	1 vs 2	1	9.514	.002
3	Cubic	28550.71	2 vs 3	1	0.318	.573

5.2 Model Parameter Estimates

Because the quadratic model fits the data best, we would interpret the relationship between the personality similarity and friendship using the estimates of the quadratic model, which are provided in Table 4.

Table 4. Parameter estimates of the quadratic model

Par	Est	Std.Error	t.value	p-value
β_{gender}	0.549	0.02	26.839	$< .001$
β_{score}	-0.111	0.013	-8.773	$< .001$
β_{class}	2.439	0.032	75.549	$< .001$
γ_1	-0.098	0.044	-2.238	.025
γ_2	0.038	0.012	3.088	.004
τ_0	0.228	0.04	5.694	$< .001$
τ_1	1.113	0.041	27.214	$< .001$
τ_2	1.720	0.043	40.097	$< .001$
τ_3	2.888	0.049	58.565	$< .001$
Residual deviance			228551.03	

By plugging the model parameter estimates into the quadratic model (Equation 7), we obtained the predicted cumulative probability for a tie to be in a category k (k=0,1,2 or 3) or below[1]. Equivalently, we can also get the probability for a tie to be in a category above k($k = 0, 1, 2,$ or 3)[2] and we will use them for the interpretation in the following.

First, all parameters are statistically significant, based on the significance level of 0.05. The coefficient of h_{gender} is 0.549. Given the levels for other covariates and latent personality distance being the same, two students of the same gender tend to have a closer relationship than otherwise, and they are less likely to have a lower-level friendship. Therefore, gender homogeneity boosts a higher level of acquaintanceship. *Second*, the coefficient h_{score} has a point estimate -0.111 (p-value$<$ 0.001). Given the same levels of other covariates and latent distance, students with more similar academic achievement (i.e., h_{score} is small) have a higher level of friendship with a greater probability than two students with some very different academic achievements. *Third*, the coefficient estimate of h_{class} is 2.439. Thus, two students from the same class are more likely to have a closer relationship. For instance, $\pi^{(4)}$ is larger for two students from the same class.

The coefficient estimate of the first-order distance (i.e., γ_1) is -0.098 (p-value =0.025) and that of the second-order distance is 0.038 (p-value= .004). For $k = 0, 1, 2,$ or 3, the quantity $\pi^{(k+1)} + \cdots + \pi^{(4)}$ is the probability for a tie to fall in a category above k. To better understand the relationship between personality similarity and friendship, we plotted these probabilities against the latent personality distance d, given two students are of the same gender (i.e., $h_{gender} = 1$), have the same academic score (i.e., $h_{score} = 0$), and are from the same class (i.e., $h_{class} = 1$). These plots are provided in Figure 5.

[1] The predicted cumulative probability is computed as

$$\begin{aligned}
p(m \in 0) &= F(0.228 - 0.549h_{gender} + 0.111h_{score} - 2.439h_{class} + 0.098d - 0.038d^2)\\
p(m \in 0,1) &= F(1.113 - 0.549h_{gender} + 0.111h_{score} - 2.439h_{class} + 0.098d - 0.038d^2)\\
p(m \in 0,1,2) &= F(1.720 - 0.549h_{gender} + 0.111h_{score} - 2.439h_{class} + 0.098d - 0.038d^2)\\
p(m \in 0,1,2,3) &= F(2.888 - 0.549h_{gender} + 0.111h_{score} - 2.439h_{class} + 0.098d - 0.038d^2).
\end{aligned}$$

[2] The probability for a time to be in a category above k ($k = 0, 1, 2,$ or 3)

$$\begin{aligned}
p(m \in 1,2,3,4) &= 1 - F(0.228 - 0.549h_{gender} + 0.111h_{score} - 2.439h_{class} + 0.098d - 0.038d^2)\\
p(m \in 2,3,4) &= 1 - F(1.113 - 0.549h_{gender} + 0.111h_{score} - 2.439h_{class} + 0.098d - 0.038d^2)\\
p(m \in 3,4) &= 1 - F(1.720 - 0.549h_{gender} + 0.111h_{score} - 2.439h_{class} + 0.098d - 0.038d^2)\\
p(m \in 4) &= 1 - F(2.888 - 0.549h_{gender} + 0.111h_{score} - 2.439h_{class} + 0.098d - 0.038d^2).
\end{aligned}$$

For a level $k = 0, 1, 2,$ or 3, the probability for tie to have a level above k is analogous to the probability of being "1" if we dichotomize the ordinal relations into binary relations at the level k.

All four probability curves are U-shapes. They decrease first and increase afterward when the latent distance increases. They reach their minimum values when the latent personality distance between the two students is 1.289. When the latent distance approaches 0, the probability for a tie in a category above k (for $k = 0, 1, 2,$ or 3) becomes larger, which indicates that the propensity for two students to have a higher level of acquaintanceship increases. Thus, similar personalities in extraversion and imagination are beneficial to the friendship between two students. When the latent personality distance is greater than 1.289, the probability for a friendship to be in a category above k increases with a larger latent personality distance. Thus, dissimilar personalities in extraversion and imagination also contribute to friendship. The results from this empirical study clearly support both "Birds of a feather flock together," and "Opposites attract."

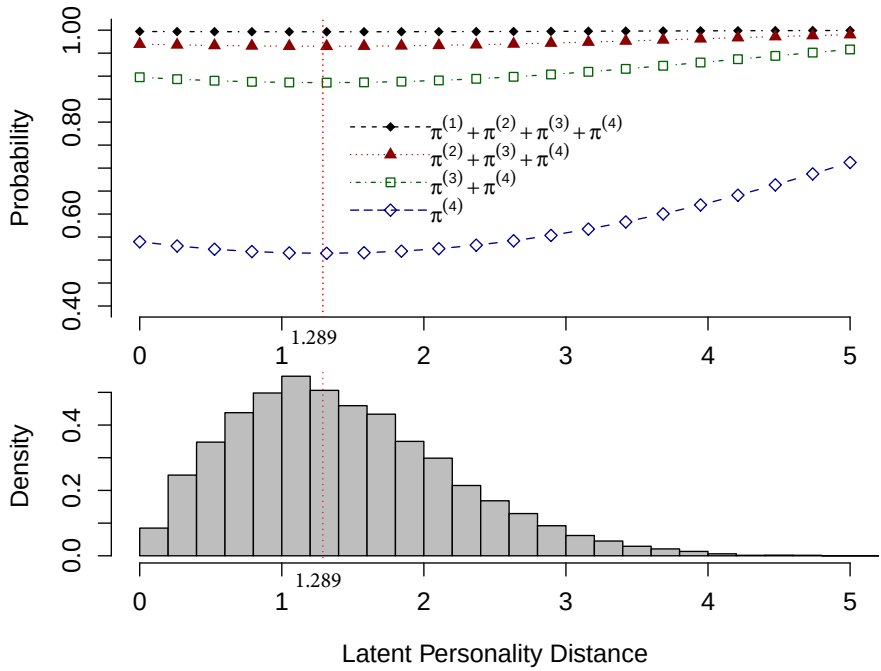


Figure 5. The top plots are the cumulative probabilities of predicted categories varying with respect to latent personality distances, and the four curves from bottom to top are the probability for a tie to be in level 4, level 3 or 4, level 2, 3, or 4, and level 1, 2, 3, or 4. The bottom panel is the density plot of personality distances; the vertical red line lies at $d = 1.289$.

6 Discussion and Conclusion

Social network analysis has been increasingly popular in recent decades. Network data are now easy to collect than ever due to the development of computer techniques. A social network comprises two primary elements: actors and potential social ties. There are observations on both dyads with social relations and dyads without social relations in social network data. Therefore, it allows researchers to understand what and how actors' characteristics predict social relations by contrasting these two groups of dyads. In the current study, we illustrated how to predict social relations using actors' characteristics by analyzing a college friendship network.

To analyze the ordinal/valued friendship network, we extended the work by Liu et al. (2018), which was built to analyze social networks with binary relations. A probit regression model was used to predict the ordinal social ties using the information of dyads. Specifically, we studied how gender homogeneity, similar academic achievements, class membership, and similar personalities predicted college student's friendship. To investigate the potential quadratic relationship between personality similarity and friendship, we fitted three competing models: a linear model with only the linear term of latent distance (i.e., d), a quadratic model with both a linear term and a quadratic term of the latent personality distance (i.e., both d and d^2), and a cubic model with also the third order of the latent personality distance. The quadratic model was significantly better than the linear model but not statistically different from the cubic model. Therefore, the quadratic models won both the linear and cubic models.

Based on the results of the quadratic model, students of the same gender or from the same class were more likely to have closer friendships. Students with similar academic scores were more likely to have higher levels of acquaintanceship. The association between personalities and friendship was mixing. Two students tended to have closer friendship relations if they had very similar personalities in extroversion and imagination. At the same time, if they were very dissimilar in those two personality traits, their friendship was more likely to fall in a higher level category. Hence, "Both birds of a feather flock together" and "Opposites attract" are possible.

Although we fitted the model for undirected networks, the modeling framework could be extended for networks with directed relations. Based on the heatmap (Figure 1), there are several communities/clusters in the college friendship network. In a cluster, students share some common characteristics. In the future, we would also like to fit multilevel models for the potential heterogeneity in the relationship between personality and friendship.

Acknowledgment

This study is partly supported by a grant from the Department of Education (R305D210023). However, the contents of the study do not necessarily represent the policy of the Department of Education, and you should not assume

endorsement by the Federal Government. We thank the reviewer and the guest editor for their instructive comments and suggestions.

References

Agresti, A. (2013). *Categorical data analysis* (3rd ed.; D. J. Balding, N. A. Cressi, G. M. Fitzmaurice, H. Goldstein, & I. M. Johnstone, Eds.). Hoboken, N.J.: John Wiley & Sons, Inc..

Altmann, T., Sierau, S., & Roth, M. (2013). I guess you're just not my type: Personality types and similarity between types as predictors of satisfaction in intimate couples. *Journal of Individual Differences*, *34*(2), 105–117. doi: https://doi.org/10.1027/1614-0001/a000105

Asendorpf, J. B., & Wilpers, S. (1998). Personality effects on social relationships. *Journal of Personality and Social Psychology*, *74*(6), 1531–1544. doi: https://doi.org/10.1037/0022-3514.74.6.1531

Bahns, A. J., Crandall, C. S., Gillath, O., & Preacher, K. J. (2017). Similarity in relationships as niche construction: Choice, stability, and influence within dyads in a free choice environment. *Journal of Personality and Social Psychology*, *112*(2), 329–355. doi: https://doi.org/10.1037/pspp0000088

Balsa, A. I., Homer, J. F., French, M. T., & Norton, E. C. (2011). Alcohol use and popularity: Social payoffs from conforming to peers' behavior. *Journal of Research on Adolescence*, *21*(3), 559–568. doi: https://doi.org/10.1111/j.1532-7795.2010.00704.x

Benenson, J. F. (1990). Gender differences in social networks. *The Journal of Early Adolescence*, *10*(4), 472–495. doi: https://doi.org/10.1177/0272431690104004

Cacioppo, J. T., & Cacioppo, S. (2014). Social relationships and health: The toxic effects of perceived social isolation. *Social and personality psychology compass*, *8*(2), 58–72. doi: https://doi.org/10.1111/spc3.12087

Cattell, R. B. (1952). *Factor analysis: an introduction and manual for the psychologist and social scientist.* Harper.

Clifton, A., & Webster, G. D. (2017). An introduction to social network analysis for personality and social psychologists. *Social Psychological and Personality Science*, *8*(4), 442–453. doi: https://doi.org/10.1177/1948550617709114

Devlieger, I., Mayer, A., & Rosseel, Y. (2016). Hypothesis testing using factor score regression: A comparison of four methods. *Educational and Psychological Measurement*, *76*(5), 741–770. doi: https://doi.org/10.1177/0013164415607618

Donnellan, M. B., Oswald, F. L., Baird, B. M., & Lucas, R. E. (2006). The mini-ipip scales: Tiny-yet-effective measures of the big five factors of personality. *Psychological assessment*, *18*(2), 192–203. doi: https://doi.org/10.1037/1040-3590.18.2.192

Elkins, L. E., & Peterson, C. (1993). Gender differences in best friendships. *Sex Roles*, *29*, 497–508. doi: https://doi.org/10.1007/BF00289323

Epskamp, S., Rhemtulla, M., & Borsboom, D. (2017). Generalized network pschometrics: Combining network and latent variable models. *Psychometrika*, *82*(4), 904–927. doi: https://doi.org/10.1007/s11336-017-9557-x

Harris, K., & Vazire, S. (2016). On friendship development and the big five personality traits. *Social and Personality Psychology Compass*, *10*(11), 647-667. doi: https://doi.org/10.1111/spc3.12287

House, J. S., Landis, K. R., & Umberson, D. (1988). Social relationships and health. *Science*, *241*(4865), 540–545. doi: https://doi.org/10.1126/science.3399889

Hudson, N. W., & Fraley, R. C. (2014). Partner similarity matters for the insecure: Attachment orientations moderate the association between similarity in partners' personality traits and relationship satisfaction. *Journal of Research in Personality*, *53*, 112–123. doi: https://doi.org/10.1016/j.jrp.2014.09.004

Jones, D. C. (1991). Friendship satisfaction and gender: An examination of sex differences in contributors to friendship satisfaction. *Journal of Social and Personal Relationships*, *8*(2), 167–185. doi: https://doi.org/10.1177/0265407591082002

Liu, H., Jin, I. H., & Zhang, Z. (2018). Structural equation modeling of social networks: Specification, estimation, and application. *Multivariate Behavioral Research*, *53*(5), 714–730. doi: https://doi.org/10.1080/00273171.2018.1479629

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, *2*, 49–55.

McCamish-Svensson, C., Samuelsson, G., Hagberg, B., Svensson, T., & Dehlin, O. (1999). Social relationships and health as predictors of life satisfaction in advanced old age: results from a swedish longitudinal study. *The International Journal of Aging and Human Development*, *48*(4), 301–324. doi: https://doi.org/10.2190/GX0K-565H-08FB-XF5G

McCrae, R. R., Martin, T. A., Hrebickova, M., Urbánek, T., Boomsma, D. I., Willemsen, G., & Costa, P. T. (2008). Personality trait similarity between spouses in four cultures. *Journal of personality*, *76*(5), 1137–1164. doi: https://doi.org/10.1111/j.1467-6494.2008.00517.x

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, *27*, 415-444. doi: https://doi.org/10.1146/annurev.soc.27.1.415

Rushton, J. P., & Bons, T. A. (2005). Mate choice and friendship in twins: evidence for genetic similarity. *Psychological Science*, *16*(7), 555–559. doi: https://doi.org/10.1111/j.0956-7976.2005.01574.x

Seeman, T. (2001). How do others get under our skin ? Social relationships and health. In C.D.Ryff & B.H.Singer (Eds.), (pp. 189–210). Oxford University Press. doi: https://doi.org/10.1093/acprof:oso/9780195145410.003.0006

Sweet, T. (2016). Social network methods for the educational and psychological sciences. *Educational Psychologist*, *51*(3-4), 381–394. doi:

https://doi.org/10.1080/00461520.2016.1208093

Thurstone, L. L. (1935). *The vectors of mind: Multiple-factor analysis for the isolation of primary traits.* University of Chicago Press.

Umberson, D., Crosnoe, R., & Reczek, C. (2010). Social relationships and health behavior across the life course. *Annual Review of Sociology*, *36*, 139–157. doi: https://doi.org/10.1146/annurev-soc-070308-120011

Waldinger, R. J., Cohen, S., Schulz, M. S., & Crowell, J. A. (2015). Security of attachment to spouses in late life: Concurrent and prospective links with cognitive and emotional well-being. *Clinical Psychological Science*, *3*(4), 516–529. doi: https://doi.org/10.1177/2167702614541261

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications* (Vol. 8). Cambridge university press.

Watson, D., Beer, A., & McDade-Montez, E. (2014). The role of active assortment in spousal similarity. *Journal of Personality*, *82*(2), 116–129. doi: https://doi.org/10.1111/jopy.12039

Watson, D., Hubbard, B., & Wiese, D. (2000). Self–other agreement in personality and affectivity: The role of acquaintanceship, trait visibility, and assumed similarity. *Journal of personality and social psychology*, *78*(3), 546–558. doi: https://doi.org/10.1037/0022-3514.78.3.546

Youyou, W., Stillwell, D., Schwartz, H. A., & Kosinski, M. (2017). Corrigendum: Birds of a feather do flock together: Behavior-based personality-assessment method reveals personality similarity among couples and friends. *Psychological Science*, *28*(3), 276-284. doi: https://doi.org/10.1177/0956797616678187

Zarbatany, L., Conley, R., & Pepper, S. (2004). Personality and gender differences in friendship needs and experiences in preadolescence and young adulthood. *International Journal of Behavioral Development*, *28*(4), 299–310. doi: https://doi.org/10.1080/01650250344000514

Journal of Behavioral Data Science, 2021, *1* (1), 53–84.
DOI: https://doi.org/10.35566/jbds/v1n1/p4

Semiparametric Bayesian Methods in Growth Curve Modeling for Nonnormal Data Analysis

Xin Tong

University of Virginia
xt8b@virginia.edu

Abstract. Semiparametric Bayesian methods have been proposed in the literature for growth curve modeling to reduce the adverse effect of having nonnormal data. The normality assumption of measurement errors in traditional growth curve models was replaced by a random distribution with Dirichlet process mixture priors. However, both the random effects and measurement errors are equally likely to be nonnormal. Therefore, in this study, three types of robust distributional growth curve models are proposed from a semiparametric Bayesian perspective, in which random coefficients or measurement errors follow either normal distributions or unknown random distributions with Dirichlet process mixture priors. Based on a Monte Carlo simulation study, we evaluate the performance of the robust models and demonstrate that selecting an appropriate model for practical data analyses is very important, by comparing the three types of robust distributional models as well as the traditional growth curve models with the normality assumption. We also provide a straightforward strategy to select the appropriate model.

Keywords: Semiparametric Bayesian methods · Growth curve modeling · Robust analysis · Dirichlet process mixture

1 Introduction

Longitudinal studies help us understand changes. Unlike one-off cross-sectional studies that give information about subjects at one point, like a snapshot photo, longitudinal studies follow subjects across time, more like a photo album. They tell a story of subjects not only at a moment in time, but also over time, showing how subjects have changed and what factors have caused between-subjects variations in change. Growth curve models are widely used in longitudinal research (e.g., McArdle & Nesselroade, 2014) as many longitudinal models in social and behavioral sciences, such as multilevel models and linear hierarchical models, can be written as a form of growth curve models. In practice, traditional growth curve model estimation is based on the assumption that both

random effects and within-subject measurement errors are normally distributed. However, data in social and behavioral sciences are rarely normal and may be contaminated by outliers (Cain et al., 2017; Micceri, 1989). Because ignoring the nonnormality of data may lead to imprecise or even inaccurate parameter estimates and misleading statistical inferences (e.g., Maronna et al., 2006; Yuan & Bentler, 2001), and routine methods, such as deleting the outliers, may lead to problems such as resulting inferences failing to reflect uncertainty and reduced efficiency (e.g., Lange et al., 1989; Yuan & Bentler, 2002), researchers have developed robust methods to obtain reliable parameter estimation and statistical inference.

The basic ideas of robust methods often include two types. The first type is to assign a weight to each subject in a dataset according to its distance from the center of the majority of the data aiming to downweight potential outlying observations (e.g., Pendergast & Broffitt, 1985; Silvapulle, 1992; Singer & Sen, 1986; Yuan & Bentler, 1998; Zhong & Yuan, 2010). The second type is to use certain nonnormal distributions that are mathematically tractable, instead of normal distributions, to model data distributions. Both types of robust methods have been directly applied to growth curve modeling. For example, on the one hand, Pendergast & Broffitt (1985) and Singer & Sen (1986) proposed robust estimators based on M-methods for growth curve models with elliptically symmetric errors, and Silvapulle (1992) further extended the M-method to allow asymmetric errors for growth curve analysis. Yuan & Zhang (2012) developed a two-stage robust procedure for structural equation modeling with nonnormal missing data and applied the procedure to growth curve modeling. On the other hand, latent variables and/or measurement errors were assumed to follow a t or skew-t distribution (Tong & Zhang, 2012; Zhang, 2016) or a mixture of certain distributions (Lu & Zhang, 2014; Muthén & Shedden, 1999). While being useful, these methods still have limitations under certain conditions. For example, the downweighting method did not perform well when latent variables contain extreme scores (e.g., see simulation results in Zhong & Yuan, 2011). Using a t distribution or a mixture of normal distributions still imposed restrictions on the shape of the data distribution.

Semiparametric Bayesian methods, also referred to as nonparametric Bayesian methods, can solve these issues as they are more flexible to relax the normality assumptions. Semiparametric Bayesian modeling relies on a building block, Dirichlet process (DP), which is a distribution over probability measures that can be used to estimate unknown distributions. Therefore, the nonnormality issue can be addressed by directly estimating the unknown random distributions of latent variables or measurement errors (i.e., obtaining the posteriors of the distributions). The advantages of using Semiparametric Bayesian methods have been discussed in the literature (e.g., Fahrmeir & Raach, 2007; Ghosal et al., 1999; Hjort, 2003; Hjort et al., 2010; MacEachern, 1999; Müller & Mitra, 2004). First, they do not constrain models to a specific parametric form that may limit the scope and type of statistical inferences in many situations. Second, they

can provide full probability models for the data-generating process and lead to analytically tractable posterior distributions.

Because of their flexibility and adaptivity, semiparametric Bayesian methods have been applied to various models. Bush & MacEachern (1996), Kleinman & Ibrahim (1998), and Brown & Ibrahim (2003) used DP mixtures to handle nonnormal random effects. Burr & Doss (2005) used a conditional DP to handle heterogeneous effect sizes in meta-analysis. Ansari & Iyengar (2006) included Dirichlet components to build a semiparametric recurrent choice model. Si & Reiter (2013) used DP mixtures of multinomial distributions for categorical data with missing values. Semiparametric Bayesian methods have also been applied to structural equation modeling to relax the normality assumption of the latent variables (e.g., Lee et al., 2008; Yang & Dunson, 2010). Tong & Zhang (2019) directly used a DP mixture to model nonnormal data in growth curve modeling. Although it has been shown in Tong & Zhang (2019) that semiparametric Bayesian methods outperformed traditional growth curve modeling as well as Student's t-distribution-based robust method when data were not normal, nonnormal data were generated with measurement errors nonnormally distributed and only measurement errors were modeled using semiparametric Bayesian methods. In practice, it is possible that random effects also violate the normality assumption. To account for this issue, we need to also model random effects semiparametrically.

Therefore, in this study, three different types of robust distributional growth curve models are proposed from a semiparametric Bayesian perspective. The features of these three types of models as well as traditional growth curve model are also discussed. In the next two sections, after introducing the idea of semiparametric Bayesian modeling, we introduce three types of semiparametric Bayesian growth curve models. Then, we compare the three types of models and the traditional model in modeling different types of data through simulation studies. Recommendations are provided at the end of the article.

2 Semiparametric Bayesian Modeling with DP Priors

A typical motivation of using semiparametric Bayesian methods is that one is unwilling to make unverified assumptions for latent variables or measurement error distributions as in the parametric modeling. Under a semiparametric perspective, we model the distribution of a random vector $\boldsymbol{\xi}$ using a random distribution function G with a prior $\mathcal{G}$. Namely, the traditional parametric assumption of the random vector $\boldsymbol{\xi}$ (i.e., $\boldsymbol{\xi} \sim N(\boldsymbol{\mu}_{\boldsymbol{\xi}}, \boldsymbol{\Phi}_{\boldsymbol{\xi}})$) is replaced by

$$\begin{aligned} \boldsymbol{\xi} &\sim G, \\ G &\sim \mathcal{G}, \end{aligned}$$

where G is an unknown distribution function and $\mathcal{G}$ is its prior, a distribution over the distribution G. The prior $\mathcal{G}$ can be chosen as the Dirichlet process (DP; Ferguson, 1973,7), which is the first prior defined for spaces of distribution

function and is the most widely used one. The Dirichlet process generates a random distribution function G, such that for any measurable partitions $P_1, \ldots, P_k$ of the sample space $\mathcal{X}$, $(G(P_1), \ldots, G(P_k))$ follows a Dirichlet distribution $Dirichlet(\alpha G_0(P_1), \ldots, \alpha G_0(P_k))$, where α and G_0 are parameters for the DP. For example, if $\mathcal{X}$ is the real space and $P = (-\infty, x]$ where x is a real number, then

$$G(x) \sim Dirichlet(\alpha G_0(x), \alpha(1 - G_0(x))).$$

Thus,

$$E(G(x)) = G_0(x),$$
$$Var(G(x)) = \frac{G_0(x)(1 - G_0(x))}{\alpha + 1}.$$

The DP is characterized by the two parameters, α and G_0. G_0 is a base distribution, which represents the central or "mean" distribution in the distribution space, while the precision parameter α governs how close realizations of G are to G_0. For example, Figure 1 displays generated random distributions from the Dirichlet process given G_0 and different values of α. The red lines in the four plots represent the cumulative density curve for the base distribution G_0, which is a standard normal distribution in this case. Black lines in each figure represent Gs generated from the Dirichlet process in five replications given G_0 and α. Clearly, as α increases, generated Gs are closer to G_0.

Ferguson (1973) introduced the DP as a random probability measure that has two desirable properties: (1) its support is sufficiently large, and (2) the posterior distribution is analytically manageable. He explained that the Dirichlet process is a conjugate prior and the posterior of G is $DP(\tilde{\alpha}, \tilde{G_0})$. The two parameters $\tilde{\alpha} = \alpha + N$ and

$$\tilde{G_0} = \frac{\alpha}{\alpha + N} G_0 + \frac{N}{\alpha + N} G_N,$$

where G_N is the empirical distribution function of the data. Thus, the posterior point estimate of G, $E(G|data) = \tilde{G_0}$, is a weighted average of two distributions: G_0 and G_N. If $\alpha = 0$, the posterior point estimate is G_N, which is nonparametric. When α approaches infinity, the posterior point estimate approaches to G_0, which is parametric. In practice, $\alpha \sim Gamma(a_1, a_2)$, which is neither 0 nor infinity. Thus, we consider the posterior point estimate of G as semiparametric.

2.1 Stick-breaking construction

Sethuraman (1994) developed a constructive way of forming G, known as "stick-breaking", and showed that draws from stick-breaking are indeed DP distributed under very general conditions. Let $q_1, q_2, \ldots, q_k, \ldots \sim Beta(1, \alpha)$. Define

$$p_k = q_k \prod_{j=1}^{k-1} (1 - q_j).$$

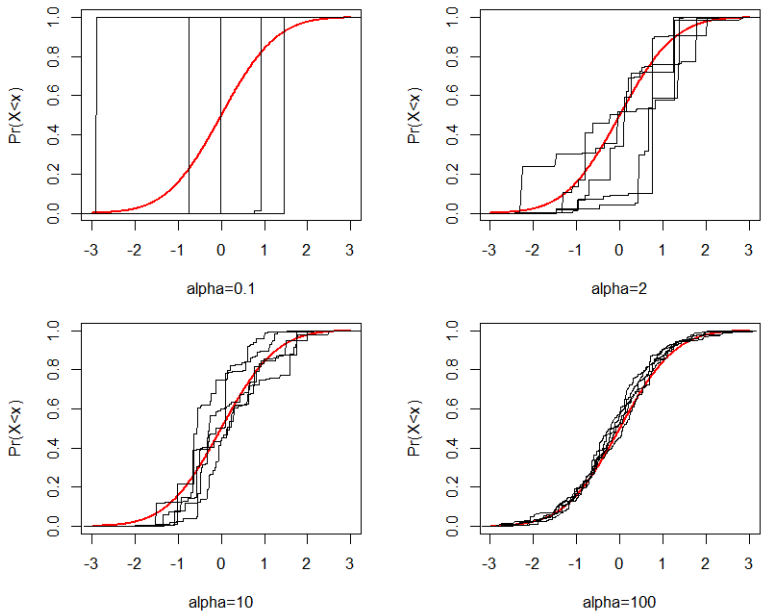


Figure 1. Random distributions generated from the Dirichlet process in five replications given a standard normal base distribution and different values of α

Then,

$$G = \sum_{k=1}^{\infty} p_k \delta_{\boldsymbol{\xi}_k^*},$$

where $\delta_{\boldsymbol{\xi}_k^*}$ is the Dirac probability measure and $\boldsymbol{\xi}_k^* \sim G_0$. It is important to note that $\sum_{k=1}^{\infty} p_k = 1$ as it guarantees G to be a distribution.

The process of the stick-breaking construction is given below.

1. Draw $\boldsymbol{\xi}_1^*$ from G_0;
2. Draw q_1 from $Beta(1, \alpha)$, then $p_1 = q_1$;
3. Draw $\boldsymbol{\xi}_2^*$ from G_0;
4. Draw q_2 from $Beta(1, \alpha)$, then $p_2 = q_2(1 - q_1)$;

...

Therefore, the distribution $G(\cdot)$ is a discrete distribution as

$$G(\cdot) = \begin{cases} \boldsymbol{\xi}_1^*, & p = p_1 \\ \boldsymbol{\xi}_2^*, & p = p_2 \\ \vdots & \vdots \\ \boldsymbol{\xi}_k^*, & p = p_k \\ \vdots & \vdots \end{cases}.$$

To define a continuous distribution, the Dirichlet process can be used as the basis of a mixture model, for example, a mixture of $N(\mu_k, \sigma_k^2)$ with mixing proportions defined by p_k. Theoretically, there are an infinite number of mixture components as $k = 1, \ldots, \infty$, given an arbitrarily flexible choice of distributional shapes. Multimodal or heavy-tailed distributions can be naturally modeled in this way. In practice, a finite number of mixture components would be good enough, and this number is taken into account by the Dirichlet process. Smaller values of DP precision parameter α result in a smaller number of mixture components.

3 Three Types of Semiparametric Bayesian Growth Curve Models

Consider a longitudinal dataset with N subjects and T measurement occasions. Let $\mathbf{y}_i = (y_{i1}, \ldots, y_{iT})'$ be a $T \times 1$ random vector with y_{ij} being an observation from subject i at time j $(i = 1, \ldots, N; j = 1, \ldots, T)$. A typical growth curve model can be written as

$$\begin{aligned} \mathbf{y}_i &= \boldsymbol{\Lambda}\mathbf{b}_i + \mathbf{e}_i, \\ \mathbf{b}_i &= \boldsymbol{\beta} + \mathbf{u}_i, \end{aligned}$$

where $\boldsymbol{\Lambda}$ is a $T \times q$ factor loading matrix that determines the growth curves, $\mathbf{b}_i$ is a $q \times 1$ vector of random effects, and $\mathbf{e}_i$ is a vector of measurement errors. The vector of random effects $\mathbf{b}_i$ varies around its mean $\boldsymbol{\beta}$. The residual vector $\mathbf{u}_i$ represents the deviation of $\mathbf{b}_i$ from $\boldsymbol{\beta}$. When

$$\boldsymbol{\Lambda} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & T-1 \end{pmatrix}, \mathbf{b}_i = \begin{pmatrix} L_i \\ S_i \end{pmatrix}, \text{ and } \boldsymbol{\beta} = \begin{pmatrix} \beta_L \\ \beta_S \end{pmatrix},$$

the model is reduced to a linear growth curve model with random intercept L_i and random slope S_i. The mean intercept and slope are denoted as β_L and β_S, respectively.

Traditionally, $\mathbf{e}_i$ and $\mathbf{u}_i$ are assumed to follow multivariate normal distributions with mean vectors of zero and covariance matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, respectively, so $\mathbf{e}_i \sim MN_T(\mathbf{0}, \boldsymbol{\Phi})$ and $\mathbf{u}_i \sim MN_q(\mathbf{0}, \boldsymbol{\Psi})$, where MN denotes

a multivariate normal distribution and its subscript indicates its dimension. Although traditional growth curve models are widely used, they can be deficient because practical data often violate the normality assumption. Tong & Zhang (2019) proposed to model $\mathbf{e}_i$ using semiparametric Bayesian methods to account for the nonnormality of data. However, since the nonnormality of a growth curve model may come from two resources – the measurement errors $\mathbf{e}_i$ and the random components $\mathbf{u}_i$ (Pinheiro et al., 2001), we model either one or both of them semiparametrically and propose three types of robust distributional growth curve models. The first type of robust semiparametric Bayesian growth curve models is the same as what Tong & Zhang (2019) proposed: we let $\mathbf{e}_i \sim G_e$, $G_e \sim DP$ and keep $\mathbf{u}_i \sim MN_q(\mathbf{0}, \boldsymbol{\Psi})$. The second type of robust growth curve models can be derived by keeping $\mathbf{e}_i \sim MN_T(0, \boldsymbol{\Phi})$ and letting $\mathbf{u}_i \sim G_u, G_u \sim DP$. The third type of robust growth curve model can be obtained by letting $\mathbf{e}_i \sim G_e$, $G_e \sim DP$ and $\mathbf{u}_i \sim G_u, G_u \sim DP$. We denote the three types of robust growth curve models as the Semi-N distributional model, the N-Semi distributional model, and the Semi-Semi distributional model, respectively. Similarly, we also denote the traditional growth curve model as the N-N distributional model.

3.1 Implementation: truncated stick-breaking construction

3.1.1 Semi-N distributional model. In the Semi-N distributional model, we assume that $\mathbf{e}_i \sim G_e$ where G_e is an unknown random distribution that is determined by the data. Because the distribution of $\mathbf{e}_i$ is continuous, a DP mixture (DPM) can be used to model the measurement errors such that

$$G_e = \begin{cases} D(\boldsymbol{\mu}_e^{(1)}, \boldsymbol{\Phi}^{(1)}), & with\ p = p_1 \\ D(\boldsymbol{\mu}_e^{(2)}, \boldsymbol{\Phi}^{(2)}), & with\ p = p_2 \\ \vdots & \vdots \\ D(\boldsymbol{\mu}_e^{(k)}, \boldsymbol{\Phi}^{(k)}), & with\ p = p_k \\ \vdots & \vdots \end{cases},$$

where D represents a predetermined multivariate distribution (e.g., multivariate normal, t, multinomial, etc.), and $\boldsymbol{\mu}_e^{(k)}$ and $\boldsymbol{\Phi}^{(k)}, k = 1, \ldots, \infty$ are means and covariances of the multivariate distribution in the kth component with probability p_k. Tong & Zhang (2019) proposed that

$$\begin{aligned} \mathbf{e}_i | \boldsymbol{\Phi}_i &\sim MN_T(\mathbf{0}, \boldsymbol{\Phi}_i), \\ \boldsymbol{\Phi}_i | G &\sim G, \\ G &\sim DP(\alpha, G_0). \end{aligned}$$

That is, the unknown distribution G_e is approximated by a mixture of multivariate normal distributions where the mixing measure has a Dirichlet process prior, $G_e \sim DPM$. The DP prior $DP(\alpha, G_0)$ can be obtained using the truncated stick-breaking construction (e.g., Lunn et al., 2013; Sethuraman,

1994). Specifically, $DP(\cdot) = \sum_{j=1}^{C} p_j \delta_{z_j}(\cdot), 1 \leq C < \infty$, where C ($1 \leq C \leq N$,often set at a large number) is a possible maximum number of mixture components, $\delta_{z_j}(\cdot)$ denotes a point mass at z_j and $z_j \sim G_0$ independently. The random weights p_j can be generated through the following procedure. With $q_1, q_2, \ldots, q_C \sim Beta(1, \alpha)$, define

$$p_j^{'} = q_j \prod_{k=1}^{j-1}(1 - q_k), j = 1, \ldots, C.$$

Then, p_j is obtained by

$$p_j = \frac{p_j^{'}}{\sum_{k=1}^{C} p_k^{'}},$$

to satisfy that $\sum_{j=1}^{C} p_j = 1$.

Thus, the distribution of $\mathbf{e}_i$ through the truncated stick-breaking construction is

$$G_e = \begin{cases} MN(\boldsymbol{\mu}_e^{(1)}, \boldsymbol{\Phi}^{(1)}), & with\ p = p_1 \\ MN(\boldsymbol{\mu}_e^{(2)}, \boldsymbol{\Phi}^{(2)}), & with\ p = p_2 \\ \vdots & \vdots \\ MN(\boldsymbol{\mu}_e^{(C)}, \boldsymbol{\Phi}^{(C)}), & with\ p = p_C \end{cases}.$$

Given that the mean of $\mathbf{e}_i$ is $\mathbf{0}$, we constrain $\sum_{j=1}^{C} p_j \boldsymbol{\mu}_e^{(j)} = \mathbf{0}$. For simplicity, we follow Tong & Zhang (2019) and constrain $\boldsymbol{\mu}_e^{(j)}$ to be 0. We use inverse Wishart priors $p(\boldsymbol{\Phi}^{(j)}) = IW(n_0, W_0)$ for the covariance matrices of the mixture components, $\boldsymbol{\Phi}^{(j)}, j = 1, \ldots, C$. Following Lunn et al. (2013, page 294), we fix the shape parameter n_0 at a specific number and assign an inverse Wishart prior to the scale matrix W_0. With such a specification, the measurement error for subject i, $\mathbf{e}_i$, has a p_j probability of coming from the mixing component $MN(\mathbf{0}, \boldsymbol{\Phi}^{(j)})$. If $\mathbf{e}_i, i = 1, \ldots, N$ are from K_e different distributions among $MN(\mathbf{0}, \boldsymbol{\Phi}^{(j)}), j = 1, \ldots, C$, K_e is called the number of clusters for $\mathbf{e}_i$. Clearly, $K_e \leq C$, and within each cluster, $\mathbf{e}_i$s come from the same distribution.

Bayesian methods are applied to estimate the model. The key idea of Bayesian methods is to compute the posterior distributions for model parameters by combining the likelihood function and the priors. Recall that in traditional N-N distributional growth curve model, $\boldsymbol{\beta}, \boldsymbol{\Phi}$, and $\boldsymbol{\Psi}$ are the model parameters. Here in the Semi-N model, $\boldsymbol{\beta}$ and $\boldsymbol{\Psi}$ are still model parameters and can be estimated in the same way. However, instead of estimating $\boldsymbol{\Phi}$ as in the N-N model, we obtain $\mathbf{e}_i$ and K_e. The estimate of K_e indicates the heterogeneity of between-subject measurement errors $\mathbf{e}_i$. With a larger value of K_e, we are more confident to conclude that different subjects' measurement errors are distributed differently. To obtain an estimate of $\boldsymbol{\Phi}$ (the covariance matrix of $\mathbf{e}_i$), we let $\mathbf{e}_{i(s)}, i = 1, \ldots, N$ be the observations of $\mathbf{e}_i$ simulated from the posterior distribution in the sth Gibbs sampler iteration, and let $\boldsymbol{\Phi}_{(s)}$ be the corresponding sample covariance matrix. An estimate of $\boldsymbol{\Phi}$ can be taken as the mean of $\boldsymbol{\Phi}_{(s)}$, averaging over all the Gibbs sampler iterations after the burn-in period.

3.1.2 N-Semi distributional model In the N-Semi model, $\mathbf{u}_i$ follow an unknown distribution G_u with a Dirichlet process prior. We can obtain the mixing proportion p_k and construct the distribution G_u in a similar way as in the Semi-N model.

$$G_u = \begin{cases} MN(\boldsymbol{\mu}_u^{(1)}, \boldsymbol{\Psi}^{(1)}), & p = p_1 \\ MN(\boldsymbol{\mu}_u^{(2)}, \boldsymbol{\Psi}^{(2)}), & p = p_2 \\ \vdots & \vdots \\ MN(\boldsymbol{\mu}_u^{(C)}, \boldsymbol{\Psi}^{(C)}), & p = p_C \end{cases},$$

where $\boldsymbol{\mu}_u^{(k)}$ and $\boldsymbol{\Psi}^{(k)}, k = 1, \ldots, C$ are parameters of the multivariate normal distribution in the kth component. Since $\mathbf{u}_i$ represents the random component of the random effects $\mathbf{b}_i$, it is also reasonable to set $\boldsymbol{\mu}_u^{(k)} = \mathbf{0}$. For the covariance matrices of the mixture components, $\boldsymbol{\Psi}^{(k)}$, inverse Wishart priors are used

$$p(\boldsymbol{\Psi}^{(k)}) = IW(m_0, V_0),$$

where m_0 and V_0 are hyperparameters.

Therefore, $\mathbf{u}_i$ comes from $MN(\mathbf{0}, \boldsymbol{\Psi}^{(k)})$ with the probability p_k. The number of clusters for $\mathbf{u}_i$ is denoted by K_u. Within each cluster, $\mathbf{u}_i$s come from the same distribution.

In contrast to the N-N and Semi-N distributional growth curve models, in the N-Semi model, we obtain $\mathbf{u}_i$ and K_u in the Markov chain Monte Carlo (MCMC) procedure instead of estimating $\boldsymbol{\Psi}$, while the fixed effects $\boldsymbol{\beta}$ and the covariance matrix of measurement errors $\boldsymbol{\Phi}$ are still model parameters and estimated in the same way. The estimate of K_u indicates the heterogeneity of random effects for different subjects. If K_u is large, we are more confident to conclude that different subjects have different growth trajectories. To obtain an estimate of $\boldsymbol{\Psi}$ (the covariance matrix of $\mathbf{u}_i$), we let $\mathbf{u}_{i(s)}, i = 1, \ldots, N$ be the observations of $\mathbf{u}_i$ simulated from the posterior distribution in the sth Gibbs sampler iteration, and let $\boldsymbol{\Psi}_{(s)}$ be the corresponding sample covariance matrix. An estimate of $\boldsymbol{\Psi}$ is the mean of $\boldsymbol{\Psi}_{(s)}$, averaging over all the Gibbs sampler iterations after the burn-in period. For the linear growth curve model, the estimate $\hat{\boldsymbol{\Psi}}$ is a 2×2 matrix $((\hat{\sigma}_L^2, \hat{\sigma}_{LS})', (\hat{\sigma}_{LS}, \hat{\sigma}_S^2)')$. The significance of $\hat{\sigma}_L^2$ and $\hat{\sigma}_S^2$ imply the existence of between-subject differences in the initial level and the rate of change, respectively. A significant $\hat{\sigma}_{LS}$ means that the initial level and the rate of change are significantly correlated.

3.1.3 Semi-Semi distributional model In the Semi-Semi model, both $\mathbf{e}_i$ and $\mathbf{u}_i$ follow unknown distributions G_e and G_u, separately. The two distributions can be constructed in the same way as in the Semi-N and N-Semi distributional models. Consequently, we cannot obtain both the estimates of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ directly, but they can be calculated following the same procedure as discussed in previous sections, and be interpreted likewise. Besides $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, other model parameters include $\boldsymbol{\beta}$, K_e, and K_u, which can be estimated

explicitly in the MCMC procedure. The fixed effect $\boldsymbol{\beta}$ represents the average initial level and rate of change for all subjects. The number of clusters for $\mathbf{e}_i$ and the number of clusters for $\mathbf{u}_i$ are K_e and K_u, indicating the heteroscedasticities of $\mathbf{e}_i$ and $\mathbf{u}_i$, respectively.

3.2 Visual model comparisons

To illustrate the differences among the N-N, Semi-N, N-Semi, and Semi-Semi distributional models, we generate and plot data from the four types of models (Figure 2). For each type of model, data on 50 subjects are generated at four occasions assuming a linear growth trend. Figure 2(a) displays the trajectories of the data generated from the N-N distributional model. No outlier can be observed. The overall trajectory looks clean and smooth. Figure 2(b) plots the data generated from the Semi-N distributional model with nonnormal measurement errors and normal random effects. Noticeably, some observations stand out of the overall trajectory such as those labeled by 1 and 2. A close look at the two observations reveals that the reason why they deviate from the overall trajectory is that they are off their own growth trajectories. Figure 2(c) portrays data from the N-Semi distributional model with normal measurement errors but nonnormal random effects. Some observations also deviate from the overall growth trajectory. However, it seems that those observations are still on their own growth trajectories. The reason why they stand out is that the rate of growth for the specific case is very different from the majority of cases. Figure 2(d) draws the trajectories for the data from the Semi-Semi distributional model with both nonnormal errors and random effects. Clearly, the outlying observations are due to two sources - the trajectory of a case deviates from the overall trajectory and the observation for this specific case is off its own trajectory. For example, observation 1 stands out because it is off the trajectory of the case and the case itself has a lower initial level and a lower rate of change. In summary, Figure 2 suggests that the four types of distributional growth curve models can imply very different patterns in growth trajectories. For instance, if a subject's growth trajectory is within the normal range of the overall trajectory and an observation at certain times stands out, the data are more likely to come from the Semi-N distributional model. If, within a subject, observations follow a smooth pattern but the trajectory itself differs from the overall trajectory, the data are more likely to come from the N-Semi distributional model. Therefore, given an empirical data set, it is very important to specify the correct type of growth curve models. In order to concretely demonstrate the possible adverse effects of misspecification for finite samples, we conduct a simulation study in the next section.

4 A Simulation Study

In this simulation study, we aim to evaluate the performance of the three robust distributional models as well as the traditional N-N model. Moreover, the effects

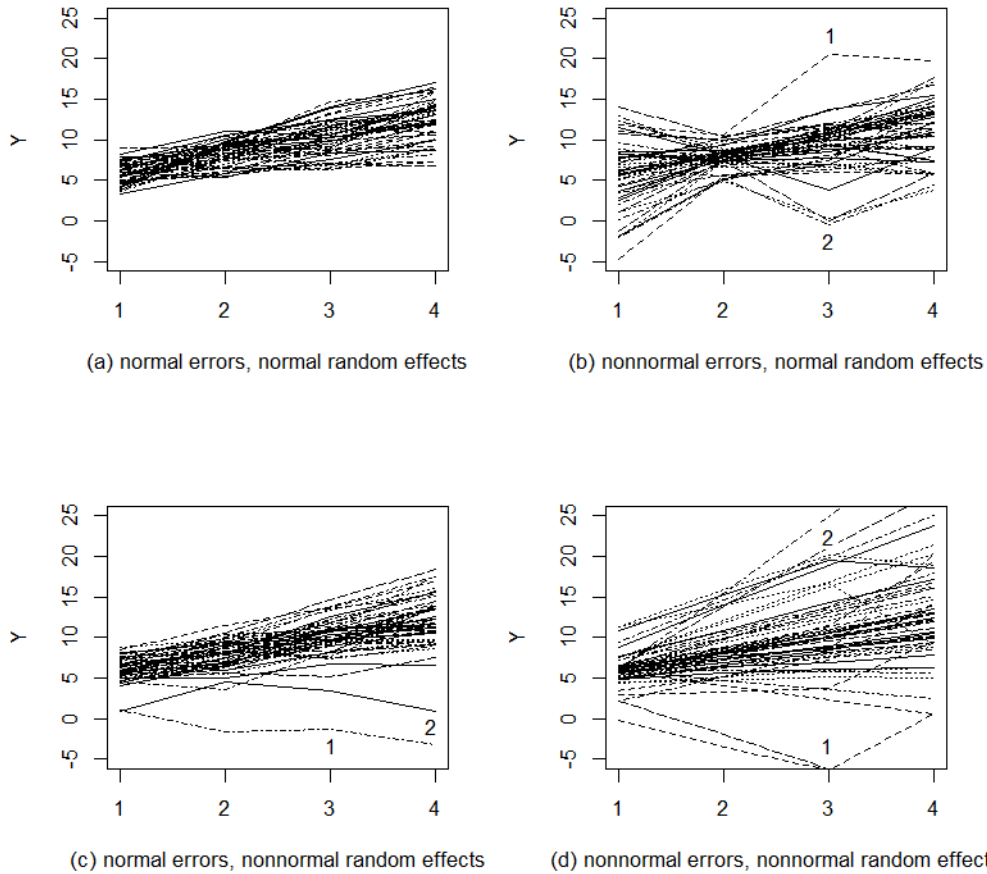


Figure 2. Trajectory plots of data generated from the 4 different types of distributional growth curve models. Data on 50 subjects are generated for 4 measurement occasions.

of the misspecification of the three types of robust distributional growth curve models will be studied to compare the intrinsic characteristics of them. We first generate data from the N-N, Semi-N, N-Semi, and Semi-Semi distributional models and name the data as N-N data, Semi-N data, N-Semi data, and Semi-Semi data, respectively. Then, for each type of data, we fit all four types of models and compare their parameter estimates.

We focus on a linear growth curve model as discussed in the previous section

$$\mathbf{y}_i = \mathbf{\Lambda b}_i + \mathbf{e}_i,$$
$$\mathbf{b}_i = \boldsymbol{\beta} + \mathbf{u}_i.$$

In the model (see Figure 3), the fixed effects are given by $\boldsymbol{\beta} = (\beta_L, \beta_S)' = (6.2, 0.3)'$.

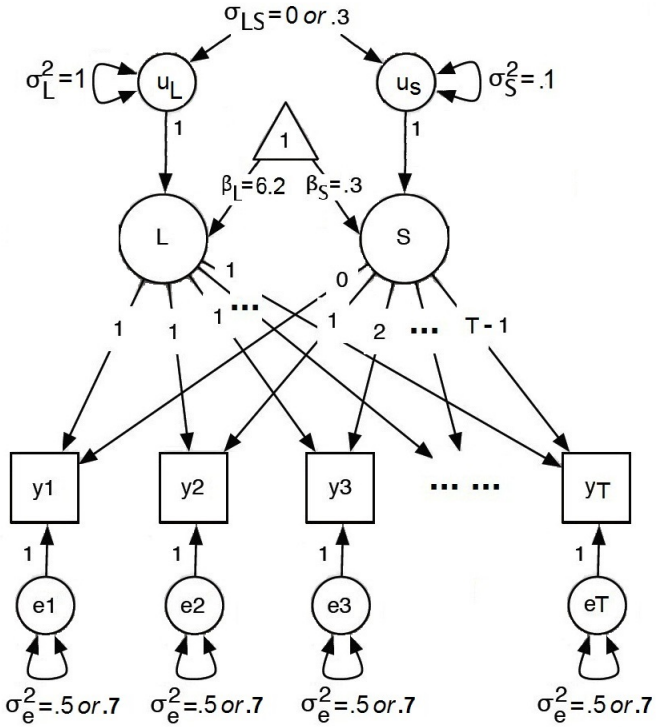


Figure 3. Path diagram of a linear growth curve model. The numbers in the path diagram are population parameter values used in the simulation.

4.1 Study design

In this study, seven possible influential factors are studied (see Table 1): type of model, type of data, potential number of clusters (C), sample size (N), number of measurement occasions (T), the covariance between the latent intercept and slope (σ_{LS}), and variance of measurement errors (σ_e^2).

First, four types of distributional growth curve models are considered, including the N-N, Semi-N, N-Semi, and Semi-Semi distributional models. Second, based on the four types of models, we generate four types of data, called N-N data, Semi-N data, N-Semi data, and Semi-Semi data correspondingly. We use each one of the four models to fit all four types of data under different conditions of the other five influential factors as described below.

(1) Three different sample sizes are considered: N =50, 200, and 500. (2) The number of measurement occasions T is either 3 or 5. (3) For the semiparametric models, we assume that data are potentially from 5 or 20 different clusters. (4) For the growth curve model parameters, the covariance between the latent intercept and the slope σ_{LS} is either 0 or 0.3, reflecting uncorrelated and correlated coefficients, respectively. When we generate $\mathbf{u}_i$ from the semiparametric perspective, we simply generate $\boldsymbol{\Psi}^{(k)} \sim IW(m_0, (m_0 - 2 - 1)\boldsymbol{\Psi})$ where $\boldsymbol{\Psi} = ((\sigma_L^2, \sigma_{LS})', (\sigma_{LS}, \sigma_S^2)')$ and the hyperparameter $m_0 = 4$ so that the mean of $\boldsymbol{\Psi}^{(k)}$ is $\boldsymbol{\Psi}$ and thus the "mean" of G_u is a distribution with its covariance matrix being $\boldsymbol{\Psi}$. (5) In practice, it is typical to assume the independence of measurement errors and the homogeneity of error variances across time, so the within-subject measurement error structure is usually simplified to $\boldsymbol{\Phi} = \sigma_e^2\mathbf{I}$. The variance of measurement errors σ_e^2 is manipulated to be 0.5 or 0.7 to investigate the influence of measurement errors. When we generate $\mathbf{e}_i = (e_{i1}, \ldots, e_{iT})'$ semiparametrically, we can set $\boldsymbol{\Phi}^{(k)} \sim IW(n_0, (n_0 - T - 1)\sigma_e^2\mathbf{I})$. However, in practice, it is easier to generate $e_{i1}, \ldots, e_{iT}$ separately from a univariate distribution $N(0, \sigma_e^{2(k)})$. We generate $\sigma_e^{2(k)}$ from $\sigma_e^{2(k)} \sim IG(c_0, d_0)$, where $c_0 = 2$ and $d_0 = \sigma_e^2$ so that the mean of $\sigma_e^{2(k)}$ is $d_0/(c_0 - 1) = \sigma_e^2$.

Overall, 768 conditions of simulations are considered. For each condition, a total of 200 data sets are generated and analyzed in OpenBUGS (Lunn et al., 2013).

4.1.1 Pseudo-procedure to generate the Semi-Semi data

1. Set C equal to the number of clusters;
2. Generate $p1_k$, $k = 1, ..., C$;
3. Generate $\sigma_e^{2(k)} \sim IG(c_0, d_0)$;
4. Generate $p2_k$, $k = 1, ..., C$;
5. Generate $\boldsymbol{\Psi}^{(k)} \sim IW(m_0, (m_0 - 2 - 1)\boldsymbol{\Psi})$;
6. For i in $1 : N$, do
   (a) Randomly select a cluster based on $p1_k$;
   (b) If the k_1th cluster is selected in (a), generate $e_{i1}, \ldots, e_{iT} \sim N(0, \sigma_e^{2(k_1)})$ and let $\mathbf{e}_i = (e_{i1}, \ldots, e_{iT})'$;
   (c) Randomly select a cluster based on $p2_k$;

Table 1. Influential factors studied in the simulation study 1

Factor	# of factor levels	Levels
Type of model	4	N-N model, Semi-N model, N-Semi model, Semi-Semi model
Type of data	4	N-N data, Semi-N data, N-Semi data, Semi-Semi data
# of clusters	3	5, 20
Sample size	2	50, 200, 500
# of measurement occasion	2	3, 5
Var(measurement errors)	2	0.5, 0.7
Cov(intercept, slope)	2	0, 0.3

(d) If the k_2th cluster is selected in (c), generate $\mathbf{u}_i \sim MN(0, \boldsymbol{\Psi}^{(k_2)})$;
(e) Generate $\mathbf{y}_i = \boldsymbol{\Lambda\beta} + \boldsymbol{\Lambda}\mathbf{u}_i + \mathbf{e}_i$.

4.2 Evaluation Criteria

We obtain the parameter estimate, bias, relative bias, empirical standard error, mean square error (MSE), and coverage probability (CP) of the 95% highest posterior density (HPD) credible intervals [1] for each parameter. Let θ denote a parameter and also its population value, and let $\hat{\theta}_r$, $r = 1, \ldots, 200$ denote its estimates from the rth simulation replication. Furthermore, let $\hat{l}_r$ and $\hat{u}_r$ denote the lower and upper limits of the 95% HPD credible interval for θ, respectively. Then, the parameter estimate of θ, $\hat{\theta}$, is calculated as the average of parameter estimates of 200 simulation replications

$$\hat{\theta} = \frac{1}{200} \sum_{r=1}^{200} \hat{\theta}_r.$$

The bias of $\hat{\theta}$ is $bias(\hat{\theta}) = \hat{\theta} - \theta$. The relative bias of $\hat{\theta}$ is

$$RB(\hat{\theta}) = \begin{cases} 100 \times \left(\dfrac{\hat{\theta}}{\theta} - 1 \right) & \theta \neq 0, \\ 100 \times \hat{\theta} & \theta = 0. \end{cases}$$

Note that the relative bias is rescaled by multiplying 100. Smaller relative bias indicates that the point estimate is less biased and thus more accurate. The empirical standard error is defined by

$$SE(\hat{\theta}) = \frac{1}{199} \sum_{r=1}^{200} \left(\hat{\theta}_r - \hat{\theta} \right)^2.$$

The mean square error is calculated by $MSE(\hat{\theta}) = bias(\hat{\theta})^2 + SE(\hat{\theta})^2$. The CP is calculated as

$$CP(\hat{\theta}) = \frac{\#(\hat{l}_r < \theta < \hat{u}_r)}{200},$$

where $\#(\hat{l}_r < \theta < \hat{u}_r)$ is the total number of replications with credible intervals covering the true parameter value θ. Good 95% HPD credible intervals should give coverage probabilities close to 0.95.

[1] Posterior credible interval, also called credible interval or Bayesian confidence interval, is analogical to the frequentist confidence interval. The 95% HPD credible interval $[l, u]$ satisfies: 1. $Prob(l \leq \theta \leq u|data) = 0.95$; 2. for $\theta_1 \in [l, u]$ and $\theta_2 \notin [l, u]$, $Prob(\theta_1|data) > Prob(\theta_2|data)$. In general, HPD intervals have the smallest volume in the parameter space of θ, and numerical methods have to be used to find HPD intervals.

4.3 Results: Part I

In this part, we evaluate the performance of the semiparametric models through comparing them with the traditional N-N model in parameter estimation.

First, when data are normally distributed, the four models perform equally well, especially for large sample sizes. For example, Table 2 contains the absolute bias and the standard errors for the six important model parameters (β_L, β_S, σ_L^2, σ_S^2, σ_{LS}, and σ_e^2) of the four distributional models, when data are generated from the N-N model with $N = 500$, $T = 5$, $C = 20$, $\sigma_{LS} = 0.3$, and $\sigma_e^2 = 0.5$. Apparently, there is no notable difference in the performance of the four models. When sample size is small, the overall pattern does not change much (see Table 3). For some parameter estimates, the semiparametric models may slightly outperform the traditional N-N model.

Table 2. Parameter estimation for the four distributional models when data are generated from the N-N model with $N = 500$, $T = 5$, $C = 20$, $\sigma_{LS} = 0.3$, and $\sigma_e^2 = 0.5$

	N-N model		Semi-N model		N-Semi model		Semi-Semi model	
	AB	SE	AB	SE	AB	SE	AB	SE
β_L	-0.004	0.049	-0.003	0.049	-0.003	0.050	-0.003	0.049
β_S	-0.002	0.017	-0.002	0.017	-0.002	0.017	-0.002	0.017
σ_L^2	0.052	0.090	0.054	0.090	0.051	0.090	0.050	0.089
σ_S^2	0.017	0.009	0.017	0.009	0.015	0.009	0.015	0.009
σ_{LS}	-0.025	0.021	-0.024	0.021	-0.026	0.021	-0.026	0.021
σ_e^2	-0.019	0.015	-0.020	0.015	-0.020	0.015	-0.020	0.015

Note. AB: absolute bias; SE: empirical standard error.

Table 3. Parameter estimation for the four distributional models when data are generated from the N-N model with $N = 50$, $T = 3$, $C = 5$, $\sigma_{LS} = 0$, and $\sigma_e^2 = 0.1$

	N-N model		Semi-N model		N-Semi model		Semi-Semi model	
	AB	SE	AB	SE	AB	SE	AB	SE
β_L	-0.001	0.157	0.004	0.161	0.001	0.158	0.001	0.158
β_S	0.007	0.053	0.005	0.054	0.006	0.053	0.006	0.054
σ_L^2	0.025	0.226	0.029	0.230	-0.016	0.221	-0.021	0.221
σ_S^2	0.039	0.028	0.037	0.027	0.019	0.028	0.018	0.028
σ_{LS}	-0.015	0.057	-0.014	0.056	-0.017	0.055	-0.015	0.055
σ_e^2	0.002	0.020	0.005	0.020	0.001	0.020	0.004	0.020

Note. AB: absolute bias; SE: empirical standard error.

Next, we evaluate the performance of the four models when data are not normally distributed. Specifically, we compare the N-N model to the Semi-N, N-Semi and Semi-Semi models in analyzing the Semi-N data, N-Semi data and

Semi-Semi data, respectively. We take a close look at the parameter estimates, bias, relative bias, empirical standard errors, MSEs, and CPs.

Table 4 contains the estimation results of the N-N and Semi-N models when $N = 200$, $T = 3$, $C = 20$, $\sigma_{LS} = 0$, and $\sigma_e^2 = 0.5$ in analyzing the Semi-N data. When data are generated with the measurement errors coming from different clusters, using the Semi-N model consistently leads to less biased estimates, smaller standard errors and MSEs, and better CPs. For the fixed effects β_L and β_S, estimates from the N-N model and the Semi-N model are about the same. Standard errors are smaller for the Semi-N model. Also, CPs of the 95% HPD credible intervals from the Semi-N model are relatively closer to the nominal level 95%. For parameters σ_L^2, σ_S^2, and σ_{LS} which are related to the random effects, the bias and standard errors are uniformly smaller by fitting the Semi-N model to the data. Furthermore, the CPs for σ_S^2 and σ_{LS} increase from 0.910 and 0.905 to 0.940 and 0.945, respectively, tending much closer to the nominal level 95%. We notice that the estimates of σ_e^2 are around 0.475 for both the N-N and Semi-N models, the standard errors are large, and the CPs are extremely different from the 95%. These are because the measurement errors e_{it} are generated from $N(0, \sigma_e^2)$, and σ_e^2 are generated from $IG(2, 0.5)$ to control the mean of σ_e^2 to be 0.5. However, data generated from $IG(2, 0.5)$ are usually less than 0.5 because this inverse Gamma distribution is skewed to the right. Therefore, in practice, we hardly can control the variance of the measurement errors when generating the Semi-N data, and thus, the bias, MSE, and CP for σ_e^2 cannot be trusted for the Semi-N data as the population parameter values are unknown. Note that the parameter estimates and their standard errors can still be trusted. For the Semi-N model, the estimated number of clusters for $\mathbf{e}_i$ is about 6 and the standard error of it is 0.653. There are 6 different clusters among the 200 subjects in the distribution of the measurement errors. Because we use informative priors for the DP precision parameter α to reduce the computational complexity and time, the estimate of α is very precise. The same pattern can be observed for all the other conditions in the comparison between the N-N and Semi-N models. Detailed tables under different conditions are available in Appendix A on our GitHub site: `https://github.com/CynthiaXinTong/SemiparametricBayeisnGCM`.

Table 5 presents the comparison between the N-N and N-Semi models when $N = 200$, $T = 5$, $C = 20$, $\sigma_{LS} = 0$, and $\sigma_e^2 = 0.1$ in analyzing the N-Semi data. The parameter estimates for the fixed effects β_L and β_S are about the same for both the N-N and N-Semi models, whereas the standard error estimates for β_L and β_S are smaller for the N-Semi model, usually resulting in smaller CPs of the HPD intervals. Under this specific condition, the CPs for the N-Semi model are closer to the nominal level 95%. For the variance estimate of the measurement error σ_e^2, fitting the two models leads to similar results as well. This phenomenon is closely related to the estimate of K_u. In this analysis, the estimate of K_u is 2.418, meaning that there are only 2 potential clusters for the random effects. In this case, using the N-Semi model may not be very different from using the traditional growth curve model. For parameter σ_L^2, σ_S^2, and σ_{LS}, their bias, MSEs, and CPs cannot be trusted. The reason is similar to the reason

Table 4. Parameter estimation for the N-N and Semi-N distributional models when data are generated from the Semi-N model with $N = 200$, $T = 3$, $C = 20$, $\sigma_{LS} = 0$, and $\sigma_e^2 = 0.5$

	N-N model						Semi-N model					
	Est.	AB	RB (%)	SE	MSE	CP	Est.	AB	RB (%)	SE	MSE	CP
β_L	6.201	0.001	0.009	0.082	0.007	0.960	6.201	0.001	0.008	0.081	0.007	0.955
β_S	0.303	0.003	0.845	0.041	0.002	0.980	0.302	0.002	0.620	0.039	0.001	0.970
σ_L^2	1.016	0.016	1.576	0.138	0.019	0.970	1.014	0.014	1.395	0.134	0.018	0.970
σ_S^2	0.135	0.035	35.280	0.035	0.002	0.910	0.132	0.032	31.663	0.028	0.002	0.940
σ_{LS}	-0.022	-0.022	-2.157	0.058	0.004	0.905	-0.019	-0.019	-1.899	0.053	0.003	0.945
σ_e^2	0.475	-0.025	-5.076	0.365	0.134	0.240	0.476	-0.024	-4.835	0.364	0.133	0.215
K_e	-	-	-	-	-	-	5.800	-	-	0.653	-	-
α	-	-	-	-	-	-	0.999	-0.001	-0.069	0.006	0.000	1.000

Note. Est.: estimate; AB: absolute bias; RB: relative bias; SE: standard error; MSE: mean square error; CP: coverage probability.

why bias, MSE, and CP cannot be trusted for parameter σ_e^2 in analyzing the Semi-N data. Here when the N-Semi data are generated, $\mathbf{u}_i$ is generated from the multivariate normal distribution $MN(\mathbf{0}, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi} = ((\sigma_L^2, \sigma_{LS})', (\sigma_{LS}, \sigma_S^2)')$ is generated from an inverse Wishart distribution $IW(4, ((1,0)', (0,0.1)'))$ to control the mean of $\boldsymbol{\Psi}$ to be $((1,0)', (0,0.1)')$. In practice, it is not possible to generate multivariate data evenly distributed around the the mean, so the population parameter values for $\boldsymbol{\Psi} = ((\sigma_L^2, \sigma_{LS})', (\sigma_{LS}, \sigma_S^2)')$ are unknown, and thus, we cannot calculate bias, MSE, and CPs for those parameters. In this analysis, we still use informative priors for the precision parameter α to reduce the computational time. The above pattern can be observed under the other conditions as well when comparing the N-N and N-Semi models (see detailed results in Appendix A on our GitHub site).

Table 5. Parameter estimation for the N-N and N-Semi distributional models when data are generated from the N-Semi model with $N = 200$, $T = 5$, $C = 20$, $\sigma_{LS} = 0$, and $\sigma_e^2 = 0.1$

	N-N model						N-Semi model					
	Est.	AB	RB (%)	SE	MSE	CP	Est.	AB	RB (%)	SE	MSE	CP
β_L	6.200	0.000	0.005	0.054	0.003	0.985	6.199	-0.001	-0.020	0.051	0.003	0.975
β_S	0.299	-0.001	-0.457	0.021	0.000	0.970	0.298	-0.002	-0.699	0.019	0.000	0.965
σ_L^2	0.836	-0.164	-16.353	1.304	1.726	0.120	0.829	-0.171	-17.113	1.299	1.715	0.050
σ_S^2	0.094	-0.006	-6.150	0.098	0.010	0.195	0.089	-0.011	-10.798	0.098	0.010	0.055
σ_{LS}	-0.009	-0.009	-0.919	0.244	0.060	0.345	-0.010	-0.010	-1.015	0.243	0.059	0.135
σ_e^2	0.099	-0.001	-0.529	0.005	0.000	0.955	0.099	-0.001	-0.737	0.005	0.000	0.950
K_u	-	-	-	-	-	-	2.418	-	-	0.789	-	-
α	-	-	-	-	-	-	0.967	-0.033	-3.309	0.008	0.001	1.000

The comparison results between the Semi-Semi and N-N models are presented in Table 6 for the Semi-Semi data when $N = 50$, $T = 3$, $C = 20$, $\sigma_{LS} = 0.3$, and $\sigma_e^2 = 0.5$. For this comparison, we can only compare the bias, standard error estimates, MSEs and CPs for the fixed effects parameters. Clearly, the absolute bias for the two models is close to each other, whereas the standard errors are consistently smaller for the Semi-Semi model than those for the traditional N-N model, indicating the efficiency of the estimates can be increased by using the robust Semi-Semi model. When generating the Semi-Semi data, we cannot manipulate the covariance matrix of $\mathbf{u}_i$ and the variance of $\mathbf{e}_i$ exactly. Therefore, the population parameter values of σ_L^2, σ_S^2, σ_{LS}, and σ_e^2 are unknown, so that the bias, MSEs, and CPs for these parameters cannot be evaluated. In Table 6, we also observe that the estimate of K_e is 4.501 and the estimate of K_u is 2.416, implying that there are about 5 clusters for $\mathbf{e}_i$ and 2 clusters for $\mathbf{u}_i$, respectively, among the 50 subjects. Different subjects' measurement errors are distributed differently, whereas their growth trajectories are not as much different. By using the informative priors for α_1 and α_2, the estimates of them are very precise. More comparison results between the Semi-Semi model and the N-N model under different conditions are available in Appendix A on `https://github.com/CynthiaXinTong/SemiparametricBayeisnGCM`.

Table 6. Parameter estimation for the N-N and Semi-Semi distributional models when data are generated from the Semi-Semi model with $N = 50$, $T = 3$, $C = 20$, $\sigma_{LS} = 0.3$, and $\sigma_e^2 = 0.5$

	N-N model						Semi-Semi model					
	Est.	AB	RB (%)	SE	MSE	CP	Est.	AB	RB (%)	SE	MSE	CP
β_L	6.195	-0.005	-0.087	0.166	0.028	0.980	6.196	-0.004	-0.060	0.147	0.021	0.970
β_S	0.300	0.000	0.161	0.079	0.006	0.980	0.298	-0.002	-0.526	0.073	0.005	0.980
σ_L^2	1.098	0.098	9.841	1.258	1.592	0.425	1.051	0.051	5.126	1.220	1.491	0.295
σ_S^2	0.247	0.147	147.283	0.300	0.112	0.710	0.217	0.117	116.946	0.151	0.037	0.635
σ_{LS}	0.157	-0.143	-47.786	0.440	0.214	0.275	0.163	-0.137	-45.702	0.351	0.142	0.165
σ_e^2	0.550	0.050	10.086	0.907	0.826	0.285	0.543	0.043	8.606	0.959	0.922	0.230
K_e	-	-	-	-	-	-	4.501	-	-	0.420	-	-
K_u	-	-	-	-	-	-	2.416	-	-	0.584	-	-
α_1	-	-	-	-	-	-	1.000	0.000	0.016	0.004	0.000	1.000
α_2	-	-	-	-	-	-	0.980	-0.020	-2.007	0.006	0.000	1.000

In sum, the performance of the four models is about the same for normally distributed data, especially when the sample size is large. When the sample size is small, even for normal data, some semiparametric models may perform slightly better than the traditional N-N model in the precision of parameter estimation. When data are not normally distributed, the traditional N-N model performs relatively worse than the semiparametric models. They may not exhibit quite different parameter estimates for fixed effects β_L and β_S, but the standard errors for all parameters are smaller for the semiparametric models than those

for the N-N model, potentially resulting in higher statistical power. In addition, the differences between the N-N model and the semiparametric models are closely related to the numbers of clusters K_e and K_u, which represents the heteroscedasticities of $\mathbf{e}_i$ and $\mathbf{u}_i$, respectively. If K_e or K_u is much larger than 1, data are more likely to be nonnormal, and the differences between the results from the N-N model and the semiparametric models should be bigger. Theoretically, if the estimates of K_e and K_u are 1, the parameter estimation from the Semi-Semi model should be the same as those from the traditional N-N model.

4.4 Results: Part II

We have shown that the semiparametric models perform at least equally well as the traditional N-N growth curve model when data are normal, and perform better when data are nonnormal. We recommend utilizing the semiparametric models in practical data analyses. Because there are three different semiparametric models, another purpose of this simulation study is to evaluate the effects of the misspecification of the three types of distributional growth curve models. Two commonly used statistics, which examine more than one performance criterion (Collins et al., 2001), are calculated for each model parameter to compare the three types of semiparametric growth curve models. The first statistic is the MSE based on 200 sets of parameter estimates and standard errors, and the second one is the CP of the 95% HPD credible intervals. The MSEs and CPs are then averaged over certain model parameters for each simulation condition. For the Semi-N data, MSEs and CPs are averaged over β_L, β_S, σ_L^2, σ_S^2, and σ_{LS}, because the MSE and CP for σ_e^2 cannot be trusted, as explained previously. For the N-Semi data, MSEs and CPs are averaged over β_L, β_S, and σ_e^2 since the population parameter values for σ_L^2, σ_S^2, and σ_{LS} are unknown. For the Semi-Semi data, MSEs and CPs are only averaged over β_L and β_S.

Table 7 summarizes the results for the analysis of each type of data by different types of distributional models with different sample sizes when $T = 5$, $C = 5$, $\sigma_{LS} = 0$, and $\sigma_e^2 = 0.1$. In the table, on the rows are the different types of generated data and on the columns are the three types of semiparametric distributional models used to analyze the generated data. In almost all situations, the model used to generate the data provides the best estimation results with smaller MSE and better credible interval coverage among the three types of robust growth curve models. For example, for the Semi-N data with $N = 200$, the Semi-N distributional model gives the best coverage probability and a comparable MSE to the other models. Similarly, for the N-Semi data with $N = 50$, the MSE for the N-Semi model is one of the smallest and the CP for the N-Semi model is the closest to the nominal level. Intuitively, we may consider the Semi-Semi model as the most general model and apply it to all the cases. However, it is not always a good idea. First, through our simulation results, although the MSEs for the Semi-Semi model are the smallest under different conditions, the CPs for the Semi-Semi model are not always the

best. By using the Semi-Semi model, the parameter estimates are slightly less accurate, while the standard errors are slightly smaller. Unexpectedly, the slight changes in the estimates and standard errors may result in a substantially lower coverage probability. Thus, the Semi-Semi distributional growth curve model is not optimal all the time. Second, theoretically, although the semiparametric approach is the same as the traditional growth curve analysis when the numbers of clusters take the value of 1, the estimated numbers of clusters are almost not possible to be 1 when we fit a semiparametric model to normal data. Because in each iteration of the MCMC sampling procedure, we count the number of clusters, which are at least 1. If in one iteration, the number of clusters happens to be bigger than 1 due to sampling errors, the estimated number of clusters cannot be exact 1. Therefore, semiparametric approach is not the same as the traditional growth curve analysis when analyzing normal data. One will lose statistical accuracy and increase type I errors by fitting the Semi-Semi distributional model to the N-N, Semi-N, or N-Semi data. Third, practically, estimating a Semi-Semi distributional model is more time-consuming than other types of models. It is often worth putting effort into determining the distributions of random effects and measurement errors to select the correct type of model.

The above results hold for different sample sizes, the number of measurement occasions, the potential number of clusters, the covariance between the latent intercept and slope, and the variance of the measurement errors. Take a closer look at the influence of these factors, we notice that the MSEs decrease as the sample size increases. By comparing Tables 7 and 8, Tables 7 and 9, Tables 7 and 10, and Tables 7 and 11, we observe separately that the number of measurement occasions, the potential number of clusters, the covariance between the latent intercept and slope, and the variance of the measurement errors do not affect the performance of the semiparametric models. More tables under different conditions are given in Appendix B on our GitHub site: `https://github.com/CynthiaXinTong/SemiparametricBayeisnGCM`.

In summary, the accuracy and efficiency of the estimation for a specific type of data closely depend on the correct specification of a model. Consequently, in practical data analyses, it is important to choose the correct type of model.

4.5 Model selection

Tong & Zhang (2012) proposed three model diagnostic methods and the "distribution checking based on individual growth curve analysis" method can be easily adopted for the semiparametric approach. In this method, an individual growth curve ($\mathbf{y}_i = \boldsymbol{\Lambda}\mathbf{b}_i + \mathbf{e}_i$) is first fitted to data from each individual. Using the least square estimation method, the individual coefficients (random effects) $\mathbf{b}_i = (b_{iL}, b_{iS})^T$ and the measurement errors $\mathbf{e}_i = (e_{i1}, \ldots, e_{iT})^T$ are estimated and retained. Let $\mathbf{b} = (\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_N)^T$ and $\mathbf{e} = (\hat{\mathbf{e}}_1, \cdots, \hat{\mathbf{e}}_N)^T$ where $\mathbf{b}$ is a $N \times 2$ matrix of individual coefficients estimates and $\mathbf{e}$ is a $N \times T$ matrix of estimated errors. Then, we test the normality of $\mathbf{e}$ and $\mathbf{b}$. If all 2 columns of $\mathbf{b}$ follow normal distributions, we consider the individual coefficients to be normally distributed. Otherwise, we consider them nonnormally distributed. Similarly, if

Table 7. Mean squared errors and coverage probabilities for different data and models ($T = 5$, $C = 5$, $\sigma_{LS} = 0$, $\sigma_e^2 = 0.1$)

		N=50			N=200			N=500		
		Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi
Semi-N	MSE	**0.016**	0.015	0.015	**0.004**	0.004	0.004	**0.002**	0.002	0.002
data	CP	**0.957**	0.672	0.674	**0.953**	0.677	0.686	**0.934**	0.642	0.642
N-Semi	MSE	0.009	**0.007**	0.007	0.003	**0.001**	0.001	0.001	**0.001**	0.001
data	CP	0.892	**0.940**	0.885	0.882	**0.943**	0.893	0.903	**0.948**	0.907
Semi-Semi	MSE	0.013	0.008	**0.008**	0.003	0.002	**0.002**	0.001	0.001	**0.001**
data	CP	0.965	0.973	**0.970**	0.968	0.975	**0.973**	0.943	0.960	**0.955**

Note. MSE: mean square error; CP: coverage probability. In the table, on the rows are the different types of generated data with sample size = 50, 200, and 500. On the columns are the three types of distributional models used to analyze the generated data. For each type of the generated data, three distributional models are fitted to them. The average MSE and CP for certain model parameters are obtained, as displayed in the table.

Table 8. Mean squared errors and coverage probabilities for different data and models ($T = 3$, $C = 5$, $\sigma_{LS} = 0$, $\sigma_e^2 = 0.1$)

		N=50			N=200			N=500		
		Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi
Semi-N	MSE	**0.014**	0.014	0.013	**0.004**	0.004	0.004	**0.002**	0.002	0.002
data	CP	**0.970**	0.803	0.816	**0.955**	0.792	0.804	**0.944**	0.794	0.799
N-Semi	MSE	0.009	**0.006**	0.006	0.006	**0.001**	0.001	0.006	**0.000**	0.000
data	CP	0.937	**0.958**	0.940	0.900	**0.933**	0.915	0.897	**0.927**	0.902
Semi-Semi	MSE	0.009	0.007	**0.007**	0.003	0.002	**0.002**	0.001	0.001	**0.001**
data	CP	0.965	0.970	**0.970**	0.968	0.975	**0.965**	0.953	0.945	**0.943**

Table 9. Mean squared errors and coverage probabilities for different data and models ($T = 5$, $C = 20$, $\sigma_{LS} = 0$, $\sigma_e^2 = 0.1$)

		N=50			N=200			N=500		
		Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi
Semi-N	MSE	**0.016**	0.015	0.015	**0.004**	0.004	0.004	**0.001**	0.001	0.001
data	CP	**0.950**	0.675	0.684	**0.947**	0.683	0.676	**0.944**	0.655	0.659
N-Semi	MSE	0.010	**0.005**	0.005	0.034	**0.001**	0.001	0.025	**0.001**	0.001
data	CP	0.903	**0.958**	0.908	0.930	**0.963**	0.927	0.878	**0.952**	0.900
Semi-Semi	MSE	0.009	0.007	**0.007**	0.003	0.002	**0.002**	0.001	0.001	**0.001**
data	CP	0.978	0.973	**0.978**	0.953	0.958	**0.960**	0.960	0.960	**0.963**

Table 10. Mean squared errors and coverage probabilities for different data and models ($T = 5$, $C = 5$, $\sigma_{LS} = 0.3$, $\sigma_e^2 = 0.1$)

		N=50			N=200			N=500		
		Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi
Semi-N	MSE	**0.017**	0.016	0.016	**0.004**	0.012	0.003	**0.002**	0.002	0.002
data	CP	**0.934**	0.598	0.594	**0.946**	0.608	0.608	**0.937**	0.587	0.587
N-Semi	MSE	0.006	**0.005**	0.005	0.002	**0.001**	0.001	0.001	**0.001**	0.001
data	CP	0.877	**0.938**	0.877	0.817	**0.913**	0.827	0.752	**0.852**	0.737
Semi-Semi	MSE	0.010	0.009	**0.008**	0.003	0.002	**0.002**	0.001	0.001	**0.001**
data	CP	0.958	0.960	**0.958**	0.948	0.950	**0.948**	0.955	0.958	**0.955**

Table 11. Mean squared errors and coverage probabilities for different data and models ($T = 5$, $C = 5$, $\sigma_{LS} = 0$, $\sigma_e^2 = 0.5$)

		N=50			N=200			N=500		
		Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi	Semi-N	N-Semi	Semi-Semi
Semi-N	MSE	**0.021**	0.021	0.020	**0.006**	0.006	0.006	**0.002**	0.002	0.002
data	CP	**0.953**	0.845	0.841	**0.939**	0.822	0.819	**0.946**	0.833	0.827
N-Semi	MSE	0.018	**0.008**	0.008	0.003	**0.002**	0.002	0.002	**0.001**	0.001
data	CP	0.870	**0.948**	0.872	0.868	**0.938**	0.865	0.872	**0.937**	0.870
Semi-Semi	MSE	0.015	0.011	**0.011**	0.004	0.003	**0.003**	0.001	0.001	**0.001**
data	CP	0.970	0.968	**0.968**	0.948	0.960	**0.953**	0.953	0.960	**0.958**

all T columns of $\mathbf{e}$ are normally distributed, the errors are viewed as from normal distributions. If $\mathbf{e}$ and $\mathbf{b}$ are not normally distributed, semiparametric approach is recommended. Based on the combination of the distributions for $\mathbf{e}$ and $\mathbf{b}$, the decision can be made according to Table 12.

Table 12. Distribution checking based on individual growth curve analysis

Errors	Individual Coefficients	Model
normal	normal	N-N distributional model
nonnormal	normal	Semi-N distributional model
normal	nonnormal	N-Semi distributional model
nonnormal	nonnormal	Semi-Semi distributional model

5 Discussion

Restricting to a parametric probability family can delude investigators and falsely make an illusion of posterior certainty (Müller & Mitra, 2004). In this study, we proposed a semiparametric Bayesian approach for growth curve analysis with nonnormal data. The normal distributions of the random effects and/or measurement errors of traditional growth curve model were replaced by random distributions with DPM priors. Thus, four types of distributional growth curve models were discussed, including the traditional N-N model, the robust Semi-N, N-Semi, and Semi-Semi models. Through a simulation study, we systematically evaluated the performance of the semiparametric Bayesian method and further assessed the effects of the misspecification of the four types of distributional growth curve models to compare the intrinsic characteristics of them. Seven potentially influential factors were considered including type of data (N-N data, Semi-N data, N-Semi data, Semi-Semi data), type of model (N-N model, Semi-N model, N-Semi model, Semi-Semi model), number of measurement occasions ($T = 3, 5$), potential number of clusters ($C = 5, 20$), the covariance between the latent intercept and slope ($\sigma_{LS} = 0, 0.3$), variance of measurement errors ($\sigma_e^2 = 0.1, 0.3$), and sample size ($N = 50, 100, 200$). Among the seven factors, the number of measurement occasions, the potential number of clusters, the covariance between the latent intercept and slope, and the variance of measurement errors were not influential to the comparison among the performance of the four types of distributional models. The following conclusions can be drawn for the other three factors.

First, the three types of semiparametric models perform as well as, or better than, the traditional N-N model, especially when data are nonnormal. When data are normally distributed, we may obtain slightly biased but more efficient parameter estimates by using the semiparametric models. It is possible for the semiparametric models to lead to worse CPs, but the MSEs are often smaller. When data are nonnormal, we recommend using the robust models instead of the traditional growth curve model as they provide much more accurate and

precise parameter estimates. Second, the semiparametric approach can improve the efficiency of the parameter estimation. For example, in Tables 4-6, the standard errors in the right panel are uniformly larger than those in the left panel, indicating the parameter estimation from the traditional growth curve analysis is less efficient. However, we would like to note that although the Semi-Semi model is the most general type of models, it is not always optimal. Misusing the Semi-Semi model could result in lower CPs and more type I errors. Moreover, fitting the Semi-Semi model to data is more time-consuming than fitting simpler models. Therefore, it is important to specify the correct type of model for practical data analyses. The "eyeball" method and the "distribution checking based on individual growth curve analysis" method can be used for model diagnostics (see Tong & Zhang, 2012). Third, the increase of the sample size can often improve the performance of all the four types of models. As shown in Tables 7-11, MSEs become smaller when sample size increases, but sample size does not affect the comparison among the four types of models. In general, we recommend using robust semiparametric models, especially when nonnormality is suspected.

For the semiparametric Bayesian approach, the normal assumption is replace by a random distribution with a DPM prior. In our study, the random distribution is a mixture of multivariate normal distributions with the mixing proportions generated following certain rules (e.g., truncated stick-breaking construction). So, similar to the finite growth mixture modeling, the number of clusters increases along with the increase of sample size. This is reasonable, because the diversity increases as more subjects are enrolled in the study. Naturally, there need to be more clusters. However, the semiparametric Bayesian growth curve modeling is different from finite growth mixture modeling. For finite growth mixture modeling, adding one additional cluster brings in several more parameters to be estimated. Thus, it is not possible to have many clusters when we conduct finite growth mixture analyses, whereas it is not a problem for us to obtain a large number of clusters if we use the semiparametric Bayesian method. The number of parameters for the semiparametric Bayesian model keeps the same no matter how many clusters there are.

We would like to note that the DP precision parameter α governs the expected number of clusters. Smaller values of α result in a smaller number of clusters. In this study, the DP precision parameter α has an informative prior $Gamma(100, 100)$ to reduce the computational complexity and convergence issue. The αs generated from the MCMC procedure are very close to 1. When α equals 1, about 90% prior weight on between 3 and 7 clusters (Lunn et al., 2013). Tong & Ke (2021) evaluated the effect of precision parameter prior on model estimation, model convergence, and computation time. They recommended using informative priors for the precision parameter, even when the information is inaccurate. Following their recommendation, the informative prior $Gamma(100, 100)$ was chosen in this study.

Limitations and future directions

In this study, we proposed to use a random mixture distribution to replace the normal assumption for robustness, but the distribution of mixture components is still specified as normal. To be more general, the distribution of mixture components can be nonnormal as well. For example, it is quite possible that the t distribution is a better substitute, and the Gamma distribution probably can better accommodate the skewness in the data. Thus, the influence of the distribution form of the mixture components needs further evaluation.

Note that we only compared the parameter estimation for model comparison. How well the models fit the data is not evaluated. Deviance Information Criterion (DIC) is widely used to evaluate the model fit in Bayesian analysis. Despite the popularity of DIC, it has received much criticism since it was proposed (Spiegelhalter et al., 2002). Celeux et al. (2006) argued that the DIC introduced by Spiegelhalter et al. for model assessment and model comparison was directly inspired by linear and generalized linear models, but it was open to different possible variations in the setting of models involving random effects, as in our robust growth curve models. A number of ways of computing DICs are proposed in Celeux et al. (2006), and their advantages and disadvantages are discussed. However, the calculation of DIC in semiparametric Bayesian analysis has not been studied. Thus, a more sophisticated way to calculate DIC should be considered deeply in the future, since DIC is an important index to evaluate the model performance.

This study focuses on robust simple linear growth curve models for demonstration. However, the same methods should work for nonlinear growth curve models as well. The performance of the more complicated semiparametric growth curve models (e.g. logistic and Gompertz models) can be studied in the future.

References

Ansari, A. & Iyengar, R. (2006). Semiparametric Thurstonian models for recurrent choices: A Bayesian analysis. *Psychometrika*, 71, 631–657. DOI: 10.1007/s11336-006-1233-5.

Brown, E. R. & Ibrahim, J. G. (2003). A Bayesian semiparametric joint hierarchical model for longitudinal and survival data. *Biometrics*, 59, 221–228. DOI: 10.1111/1541-0420.00028.

Burr, D. & Doss, H. (2005). A Bayesian semiparametric model for random-effects meta-analysis. *Journal of the American Statistical Association*, 100, 242–251. DOI: 10.1198/016214504000001024.

Bush, C. A. & MacEachern, S. N. (1996). A semiparametric Bayesian model for randomised block designs. *Biometrika*, 83, 275–285. DOI: 10.1093/biomet/83.2.275.

Cain, M. K., Zhang, Z., & Yuan, K.-H. (2017). Univariate and multivariate skewness and kurtosis for measuring nonnormality: Prevalence, influence

and estimation. *Behavior Research Methods*, 49, 1716–1735. DOI: 10.3758/s13428-016-0814-1.

Celeux, G., Forbes, F., Robert, C. P., & Titterington, D. M. (2006). Deviance information criteria for missing data models. *Bayesian Analysis*, 1, 651–673. DOI: 10.1214/06-ba122.

Collins, L., Schafer, J., & Kam, C. (2001). A comparison of inclusive and restrictive missing-data strategies in modern missing-data procedures. *Psychological Methods*, 6, 330–351.

Fahrmeir, L. & Raach, A. (2007). A Bayesian semiparametric latent variable model for mixed responses. *Psychometrika*, 72, 327–346. DOI: 10.1007/s11336-007-9010-7.

Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1, 209–230. DOI: 10.1214/aos/1176342360.

Ferguson, T. (1974). Prior distributions on spaces of probability measures. *The Annals of Statistics*, 2, 615–629. DOI: 10.1214/aos/1176342752.

Ghosal, S., Ghosh, J., & Ramamoorthi, R. (1999). Posterior consistency of Dirichlet mixtures in density estimation. *The Annals of Statistics*, 27, 143–158. DOI: 10.1214/aos/1018031105.

Hjort, N. L. (2003). Topics in nonparametric Bayesian statistics. In P. Green, N. L. Hjort, & S. Richardson (Eds.), *Highly Structured Stochastic Systems* (pp. 455–487). Oxford: Oxford University Press.

Hjort, N. L., Holmes, C., Müller, P., & Walker, S. G. (2010). *Bayesian nonparametrics*. Cambridge: Cambridge University Press. DOI: 10.1093/acprof:oso/9780199695607.003.0013.

Kleinman, K. P. & Ibrahim, J. G. (1998). A semiparametric Bayesian approach to the random effects model. *Biometrics*, 54, 921–938. DOI: 10.2307/2533846.

Lange, K. L., Little, R. J. A., & Taylor, J. M. G. (1989). Robust statistical modeling uisng the t distribution. *Journal of the Americal Statistical Association*, 84(408), 881–896.

Lee, S. Y., Lu, B., & Song, X. Y. (2008). Semiparametric Bayesian analysis of structural equation models with fixed covariates. *Statistics in Medicine*, 27, 2341–2360. DOI: 10.1002/sim.3098.

Lu, Z. & Zhang, Z. (2014). Robust growth mixture models with non-ignorable missingness: Models, estimation, selection, and application. *Computational Statistics and Data Analysis*, 71, 220–240. DOI: 10.1016/j.csda.2013.07.036.

Lunn, D., Jackson, C., Best, N., Thomas, A., & Spiegelhalter, D. (2013). *The BUGS book: A practical introduction to Bayesian analysis*. Boca Raton, FL: CRC Press.

MacEachern, S. (1999). Dependent nonparametric processes. In A. S. Association (Ed.), *ASA Proceedings of the Section on Bayesian Statistical Science*.

Maronna, R. A., Martin, R. D., & Yohai, V. J. (2006). *Robust statistics: Theory and methods*. New York: John Wiley & Sons, Inc. DOI:

10.1002/0470010940.

McArdle, J. J. & Nesselroade, J. R. (2014). *Longitudinal data analysis using structural equation models*. American Psychological Association. DOI: 10.1037/14440-000.

Micceri, T. (1989). The unicorn, the normal curve, and other improbable creatures. *Psychological Bulletin*, 105, 156–166. DOI: 10.1037/0033-2909.105.1.156.

Müller, P. & Mitra, R. (2004). Bayesian nonparametric inference - why and how. *Bayesian Analysis*, 1, 1–33. DOI: 10.1214/13-ba811.

Muthén, B. & Shedden, K. (1999). Finite mixture modeling with mixture outcomes using the EM algorithm. *Biometrics*, 55, 463–469. DOI: 10.1111/j.0006-341X.1999.00463.x.

Pendergast, J. F. & Broffitt, J. D. (1985). Robust estimation in growth curve models. *Communications in Statistics: Theory and Methods*, 14, 1919–1939. DOI: 10.1080/03610928508829021.

Pinheiro, J. C., Liu, C., & Wu, Y. N. (2001). Efficient algorithms for robust estimation in linear mixed-effects models using the multivariate t distribution. *Journal of Computational and Graphical Statistics*, 10(2), 249–276.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4, 639–650.

Si, Y. & Reiter, J. P. (2013). Nonparametric Bayesian multiple imputation for incomplete categorical variables in large-scale assessment surveys. *Journal of Educational and Behavioral Statistics*, 38, 499–521. DOI: 10.3102/1076998613480394.

Silvapulle, M. J. (1992). On M-methods in growth curve analysis with asymmetric errors. *Journal of Statistical Planning and Inference*, 32, 303–309. DOI: 10.1016/0378-3758(92)90013-i.

Singer, J. M. & Sen, P. K. (1986). M-methods in growth curve analysis. *Journal of Statistical Planning and Inference*, 13, 251–261. DOI: 10.1016/0378-3758(86)90137-0.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Linde, A. v. d. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4), 583–639.

Tong, X. & Ke, Z. (2021). Assessing the impact of precision parameter prior in Bayesian nonparametric growth curve modeling. *Frontiers in Psychology*. DOI: 10.3389/fpsyg.2021.624588.

Tong, X. & Zhang, Z. (2012). Diagnostics of robust growth curve modeling using Student's t distribution. *Multivariate Behavioral Research*, 47, 493–518. DOI: 10.1080/00273171.2012.692614.

Tong, X. & Zhang, Z. (2019). Robust Bayesian approaches in growth curve modeling: Using Student's t distributions versus a semiparametric method. *Structural Equation Modeling: A Multidisciplinary Journal*, 27, 544–560. DOI: 10.1080/10705511.2019.1683014.

Yang, M. & Dunson, D. B. (2010). Bayesian semiparametric structural equation models with latent variables. *Psychometrika*, 75, 675–693. DOI: 10.1007/s11336-010-9174-4.

Yuan, K.-H. & Bentler, P. M. (1998). Structural equation modeling with robust covariances. *Sociological Methodology*, 28, 363–396. DOI: 10.1111/0081-1750.00052.

Yuan, K.-H. & Bentler, P. M. (2001). Effect of outliers on estimators and tests in covariance structure analysis. *British Journal of Mathematical and Statistical Psychology*, 54, 161–175. DOI: 10.1348/000711001159366.

Yuan, K.-H. & Bentler, P. M. (2002). On normal theory based inference for multilevel models with distributional violations. *Pschometrika*, 67, 539–561.

Yuan, K.-H. & Zhang, Z. (2012). Robust structural equation modeling with missing data and auxiliary variables. *Psychometrika*, 77, 803–826.

Zhang, Z. (2016). Modeling error distributions of growth curve models through Bayesian methods. *Behavior Research Methods*, 48, 427–444. DOI: 10.3758/s13428-015-0589-9.

Zhong, X. & Yuan, K.-H. (2010). Weights. In N. J. Salkind (Ed.), *Encyclopedia of research design* (pp. 1617–1620). Thousand Oaks, CA: Sage. DOI: 10.4135/9781412961288.

Zhong, X. & Yuan, K.-H. (2011). Bias and efficiency in structural equation modeling: Maximum likelihood versus robust methods. *Multivariate Behavioral Research*, 46, 229–265. DOI: 10.1080/00273171.2011.558736.

Journal of Behavioral Data Science, 2021, *1* (1), 85–126.
DOI: https://doi.org/10.35566/jbds/v1n1/p5

Factor or Network Model? Predictions From Neural Networks

Alexander P. Christensen[1] and Hudson Golino[2]

[1] University of Pennsylvania
alexpaulchristensen@gmail.com
[2] University of Virginia
hfg9s@virginia.edu

Abstract. The nature of associations between variables is important for constructing theory about psychological phenomena. In the last decade, this topic has received renewed interest with the introduction of psychometric network models. In psychology, network models are often contrasted with latent variable (e.g., factor) models. Recent research has shown that differences between the two tend to be more substantive than statistical. One recently developed algorithm called the *Loadings Comparison Test* (LCT) was developed to predict whether data were generated from a factor or small-world network model. A significant limitation of the current LCT implementation is that it's based on heuristics that were derived from descriptive statistics. In the present study, we used artificial neural networks to replace these heuristics and develop a more robust and generalizable algorithm. We performed a Monte Carlo simulation study that compared neural networks to the original LCT algorithm as well as logistic regression models that were trained on the same data. We found that the neural networks performed as well as or better than both methods for predicting whether data were generated from a factor, small-world network, or random network model. Although the neural networks were trained on small-world networks, we show that they can reliably predict the data-generating model of random networks, demonstrating generalizability beyond the trained data. We echo the call for more formal theories about the relations between variables and discuss the role of the LCT in this process.

Keywords: neural networks · machine learning · data generating mechanisms

The nature of associations between observable variables is one of the most critical considerations for constructing theory about psychological phenomena (Borsboom, van der Maas, Dalege, Kievit, & Haig, 2020; Haslbeck, Ryan, Robinaugh, Waldorp, & Borsboom, 2019). Whether variables are associated because they all have a common cause or because they reciprocally cause and effect one

another is (ideally) theorized by the researcher and (often) implied by their choice of psychometric model (Borsboom, 2006; Haslbeck, Ryan, Robinaugh, Waldorp, & Borsboom, 2019). Determining whether empirical data are generated by one of these mechanisms is therefore an important question (van Bork et al., 2019). Although other possibilities exist (Kruis & Maris, 2016; Marsman et al., 2018), these two explanations are perhaps the most common perspectives in psychology and correspond to latent variable and network models, respectively. The debate over the plausibility of these mechanisms has sparked renewed interest in the ontology and epistemology of psychological phenomena (Borsboom, 2008; Guyon, Falissard, & Kop, 2017).

Factor (latent variable) models are represented by arrows going from latent (unobservable) variables to observable variables. From a causal theory perspective, this representation suggests that a factor causes the response to the observable variables (Edwards & Bagozzi, 2000). Network models represent observable variables as nodes (circles) and their relationships (e.g., partial correlations) as edges (lines). From a causal theory perspective, this representation suggests that observed variables directly and reciprocally cause one another (van der Maas et al., 2006). For both models, researchers may not interpret the models causally but instead as summaries of covariance. In the last few years, the apparent differences between these models have been shown to be more substantive than statistical (Guyon, Falissard, & Kop, 2017), with several studies demonstrating that both models can produce similar covariance patterns and model parameters (e.g., dimensions and loadings; Golino & Epskamp, 2017; Hallquist, Wright, & Molenaar, 2019; Marsman et al., 2018; van Bork et al., 2019; Waldorp & Marsman, 2020).

Recent simulation studies, for example, have demonstrated that clusters of nodes in networks identified by community detection algorithms (Fortunato, 2010) are consistent with latent factors in factor models (Christensen, Garrido, & Golino, 2021; Golino & Epskamp, 2017; Golino et al., 2020). Other simulations have demonstrated that *node strength* or the absolute sum of a node's connections in a network is consistent with confirmatory (Hallquist, Wright, & Molenaar, 2019) and exploratory factor loadings (Christensen & Golino, 2021). Despite producing similar model parameters, the substantive interpretations and representations of these models imply different data generating mechanisms. The implications of these different data generating mechanisms are important: Should a researcher use factor or network analysis to model their data? More significantly, should clinicians treat an underlying psychopathological disorder (factor model) or the symptoms that constitute the disorder (network model; Borsboom, 2017)?

To answer these questions, the present research aimed to develop an algorithm that could determine whether data were generated from a factor or network model. Such a tool allows researchers to determine whether their data are structured more like their hypothesized data generating mechanism. Although data generated from either model can fit and be represented by the other (van Bork et al., 2019; van der Maas et al., 2006), researchers should attempt to design, use,

and model measures that align with their theoretical perspective (Christensen, Golino, & Silvia, 2020). Recent developments have demonstrated that factor and network models can potentially be distinguished by correlation patterns of the data (Christensen & Golino, 2021; van Bork et al., 2019). One of these methods, called the *Loadings Comparison Test* (LCT), compares loadings from factor and network models to predict the data-generating model (Christensen & Golino, 2021). In its current form, however, the LCT relies on descriptive heuristics, which are unlikely to generalize across many data conditions. To make the algorithm more robust, we used artificial neural networks from machine learning. We then performed a simulation to evaluate whether the neural networks perform better than the original heuristic-based algorithm and a set of regularized logistic regression models.

1 Loadings Comparison Test

The LCT was inspired by van Bork et al. (2019) who demonstrated that unidimensional factor models and sparse network models have subtle statistical differences that can be used to determine whether the empirical data are more likely generated from one model or the other. In their paper, they identified two key differences: (1) the proportion of partial correlations that have a different sign than the corresponding zero-order correlations and (2) the proportion of partial correlations that are stronger than the corresponding zero-order correlations. The empirical value of these proportions is then compared against the distributions of data generated from factor and network models applied to simulated covariance matrices. The model with the greater probability is determined to be the most likely model. They referred to this test as the *Partial Correlation Likelihood Test*.

The Partial Correlation Likelihood Test provides a test for determining whether data are more likely generated from a factor or network model in unidimensional data structures. Although unidimensional structures are critical to psychology, the Partial Correlation Likelihood Test may not generalize to more complex models (e.g., multidimensional models; van Bork et al., 2019). The LCT was motivated by the need for such a test in multidimensional data. The development of the LCT was based on the descriptive differences between factor and network loadings when data were factor or network model (Christensen & Golino, 2021). Network loadings are the standardized sum of each node's connections to nodes in each community in a network. Below, we provide notation for how network loadings are computed.

Let $\mathbf{W}$ represent a symmetric $n \times n$ partial correlation network matrix where n is the number of nodes. Node strength is defined as:

$$S_i = \sum_{j=1}^{n} |\mathbf{W}_{ij}|$$

where $|\mathbf{W}_{ij}|$ is the absolute edge weight between node i and j and S_i is node strength for node i. Using this definition, node strength can be split by communities estimated in the network:

$$\ell_{ic} = \sum_{j \in c}^{C} |\mathbf{W}_{ij}|,$$

where ℓ_{ic} is the sum of the edge weights in community c that are connected to node i and C is the number of communities in the network. ℓ_{ic} can be standardized using:

$$\aleph_{ic} = \frac{\ell_{ic}}{\sqrt{\sum \ell_c}},$$

where $\sqrt{\sum \ell_c}$ is the square root of the sum of all edge weights for nodes in community c and $\aleph_{ic}$ is the standardized network loading for node i in community c. Signs are added after the loadings have been computed following the same procedure as factor loadings (Comrey & Lee, 2013).

Across three simulations, Christensen and Golino (2021) demonstrated that factor and network loadings are roughly equivalent when data are generated by a factor model. To determine whether this equivalency held across other data generating mechanisms, they generated data from random correlation matrices with small correlations (between ±.15) and small-world networks. They found that factor and network loadings were no longer consistent with one another when data were generated from either data generating method. This observation led them to develop a heuristic-based algorithm (LCT) that could potentially be used to determine the data generating mechanism.

1.1 Original Algorithm

The algorithm starts by generating data from a multivariate normal distribution based on the empirical covariance matrix and estimating the number of communities (or dimensions) using exploratory graph analysis (EGA; Golino & Epskamp, 2017; Golino et al., 2020). EGA estimates a network and then applies the Walktrap community detection algorithm (Pons & Latapy, 2006) to identify the number of communities in the network (see Appendix A.1 for statistical details). Using the number of dimensions estimated by EGA, factor loadings are computed using EFA with oblimin rotation. Similarly, network loadings are computed with the EGA results. This process is repeated 100 times and loadings are computed for each generated dataset.

Next, the proportions of loadings that are greater than or equal to small, moderate, and large effect sizes are computed. For factor models, these effect sizes are 0.40, 0.55, and 0.70, respectively (Comrey & Lee, 2013). For network models, these effect sizes are 0.15, 0.25, and 0.35, respectively (Christensen & Golino, 2021). Dominant and cross-loadings that are greater than or equal to small effect sizes are also computed. The proportion of loading effect sizes are computed to

summarize the covariance matrix into the same dimensions no matter how many variables are in a dataset. More specifically, any $n \times n$ covariance matrix can be summarized by these five loadings proportions for both factor and network loadings, resulting in a comparable structure (ten loading proportions in total) for all datasets.

We summarize Christensen and Golino's (2021) rationale for why there might be differences between factor and network models. Factor loadings are derived by extracting the common covariance between variables. This computation of factor loadings means that the magnitude of factor loadings depend on the shared variance across sets of variables. In contrast, network loadings are computed using the standardized sum of each node's connection to nodes in a certain dimension. This computation means that their magnitudes only depend on the covariance of each node with other nodes in a dimension. When data are generated from a factor model, then there is usually common covariance to extract in each dimension. This common covariance leads factor and network loadings to be consistent with one another as Christensen and Golino (2021) demonstrate.

Data generated from network models, however, do not imply common covariance in each dimension but rather each node usually represents its own dimension (Cramer et al., 2012). Many real-world networks tend to have a small-world structure (e.g., psychopathological disorders; Borsboom, Cramer, Schmittmann, Epskamp, & Waldorp, 2011), which are characterized by nodes having many neighboring connections but also some cross-network connections with even fewer nodes that act as hubs or nodes with an above average number of connections (Watts & Strogatz, 1998). This structure suggests that there might be common covariance between variables, but they are not necessarily structured in a systematic way—that is, common covariance is not necessarily structured in well-defined dimensions like factor models. Such a structure suggests that common covariance may be identified across some variables but will be relatively diffuse in general (i.e., across factors; Christensen & Golino, 2021). In contrast, network loadings partition the covariance based on the dimension structure (rather than common covariance), leading to a greater prevalence of loadings that are likely to be at least small or larger. Finally, network loadings would also be expected to have greater proportions of cross-loadings due to the partitioning, rather than extraction, of common covariance. These differences between the two loadings may thus be informative for determining whether data were generated from a factor or network model.

The heuristics of the LCT algorithm were developed in part based on this empirical rationale as well as simulated data. The first heuristic is the ratio of small effect size (or larger) network loadings divided by small effect size (or larger) factor loadings. When this ratio is greater than 1.5, then the algorithm suggests the data are generated from a network model; if not, a second heuristic is applied. The second heuristic is the logarithm of the ratio of dominant factor loadings that are a small effect size (or larger) divided by cross-factor loadings that are a small effect size (or larger). When this logarithm ratio is greater than 5, then the algorithm suggests the data are generated from a factor model; otherwise,

a network model. This latter heuristic was derived post-hoc for simulated data with large correlations between factors (0.70). Although simple, these heuristics performed remarkably well in simulated samples: 77.9% to 100% accuracy for factor models and 87.8% to 95.8% accuracy for network models (Christensen & Golino, 2021).

Despite high accuracy for all models, there were a couple limitations in their validation. First, sample sizes were all generated with 1000 cases, which is large relative to many samples used in psychology. Second, the simulated models used novel data but with the same data structures that the heuristics were derived from. The number of variables, for example, was held constant at fifteen for all models, and factor models were always generated with three factors and five variables per factor. These limitations are likely to result in overfitting and a lack of generalizability to other samples and data structures. These limitations motivated the current study where we sought to improve the LCT algorithm by replacing these simple heuristics with a more sophisticated computational approach: artificial neural networks.

2 Artificial Neural Networks

Artificial neural networks are a commonly used technique in machine learning research (Dreiseitl & Ohno-Machado, 2002). They come in many forms but perhaps the most basic are feed-forward networks where data are input as nodes and are "fed through" the network to output nodes (i.e., the prediction). In machine learning terms, neural networks are a supervised learning model, which means that the researcher supplies both the input variables and the output variables that the neural network must then "learn" a mapping between them. In our study, the input corresponded to the factor and network loading proportions. The output corresponded to the data-generating model (either factor or network). The mapping between the input and output occurs through the *hidden layers* of the neural network where the model learns the appropriate weighting scheme that optimizes the prediction of the output from the input.

A neural network with no hidden layers can represent linear functions only and is equivalent to a standard regression model (e.g., an output node with a sigmoid activation function is a logistic regression model). With a single hidden layer, a neural network can approximate "any function that has a continuous mapping from one finite space to another" (Heaton, 2008). Two hidden layers can represent any arbitrary boundary (e.g., non-linear functions), approximating any mapping between the input and output (Hornik, 1991; Sontag, 1991). Key to training neural networks is deciding on the number of hidden layers and the number of neurons (or nodes) in each of the hidden layers. More complex mappings require more complex neural networks (i.e., more nodes and layers).

An important concept for neural network learning is *backpropogation*. Backpropogation refers to the adjustment of weights and biases in the network (starting from the output *back* to the input; Watt, Borhani, & Katsaggelos, 2016). In training, *batches* or a certain number of samples of the data are fed through the

network's weights and predictions are made about the output. With each batch, the network updates its weights and biases by trying to minimize the loss of information between the predicted output and the actual output. The end goal is to minimize the loss of information between the predicted and actual output to maximize the accuracy of the neural network's predictions.

One of the advantages of neural networks is that they can learn mappings between the input and output that are otherwise difficult to abstract (e.g., non-linear relationships). In our case, going beyond simple descriptive heuristics to map loading proportions to the data-generating model. This advantage of neural networks is also a disadvantage. The mapping is often a "black box" that does not offer clear interpretations of the underlying function—that is, what exactly the neural network is using to distinguish a factor model from a network model.

2.1 Training the Neural Networks

In this section, we briefly describe the training procedure we used to arrive at our final neural networks (a full description of the training process can be found in Appendix A.2). Based on the original LCT algorithm, we expected certain conditions to be more difficult to predict the data-generating model. Specifically, we expected the size of the correlation between factors to have a substantial effect on prediction accuracy. To this end, we started by training two neural network models: one with low correlations between factors (0.00 and 0.30) and another with high correlations between factors (0.50 and 0.70). Such a strategy is often referred to as an *ensemble* of networks (Zhou, Wu, & Tang, 2002) where each network is fine-tuned to a specific part of the problem to improve the overall prediction of a more complex problem. The rationale for building several neural networks to predict different factor models from network models is that different information is likely to be more relevant for one set of factor models than another (primarily along the lines of the magnitude of correlations between factors).

During the training of neural networks, part of the data is "held out" from the network's learning. Consistent with the literature, we used an 80/20 split of our data where 80% of the data is used to train the network and 20% of the data is used to validate the training. The purpose of this procedure is to evaluate the neural network on data that was not used in its training. During this procedure, we found that the high correlations between factors neural network was not very accurate. We discovered that there were specific conditions where the neural network was unable to predict the data-generating model. These conditions were where the number of variables per factor was greater than the number of factors. Based on this finding, we used two neural networks for factor models with high correlations between factors (0.50 and 0.70): one with the number of variables per factor greater than the number of factors and another with the number of variables per factor less than the number of factors. The training validation accuracy of both neural networks was sufficient.

Our final neural network ensemble consisted of three neural networks: low correlations between factors, high correlations between factors with the number of variables per factor greater than the number of factors, and high correlations

between factors with the number of variables per factor less than the number of factors. Our ensemble worked by having each neural network make a prediction for whether the data were generated from a factor or network model. If *any* of the neural networks predicted a factor model, then the ensemble suggests a factor model. Conversely, if all neural networks predicted a network model, then the ensemble suggests a network model. To determine whether a neural network approach was necessary, we compared their performance to corresponding logistic regression models that were regularized using the least absolute shrinkage and selection operator (LASSO; Tibshirani, 1996). Logistic regression is commonly used as a comparison method and is useful for determining the expected baseline performance of a neural network (Dreiseitl & Ohno-Machado, 2002).

3 Present Study

In our present study, we set out to validate the neural networks against Christensen and Golino's (2021) original LCT heuristics and the logistic regression models that were trained alongside the neural networks. Although the neural networks were already validated on novel samples held out from their training samples, we sought to further test their generalizability by generating data using different conditions than the ones they were trained on—that is, manipulating the parameters of the factor and network models such that they were novel. Further, we generated data from random network models, which were not used to train the neural network and logistic regression models or the development of the original heuristic-based algorithm. Random network models are generated by a random process, making dependencies between variables unsystematic. Because the random network models are completely novel, they represent an ideal test of generalizability.

The original algorithm relied on a bootstrap approach (e.g., generating 100 samples) to compute the loadings proportion heuristics used to predict the model. In contrast, the neural network and logistic regression approaches can make predictions based on the empirical data. One potential advantage of the neural network and logistic regression approaches is that they can also be applied to each sample of the bootstrap data. Beyond the empirical predictions, the means of the loadings proportions could be computed and used to make a prediction. Another prediction could made based on the proportion of each time a model was predicted from the data (e.g., more than 50% of the samples suggesting a model predicts that model). In our simulation, we tested each type of prediction (hereafter referred to as *empirical*, *bootstrap*, and *proportion*, respectively).

4 Methods

4.1 Data Generation

All data were generated as continuous variables and sample sizes for all models were generated with 400 and 750 cases. For each model, a total of 7,200 samples

were generated, resulting in 21,600 total samples. Conditions of each model consisted of different parameter settings than what the neural network and logistic regression models were trained on. The random network models were completely novel to all LCT configurations.

4.1.1 Factor model We generated data from multivariate normal factor models following the same approach as Golino et al. (2020). First, the reproduced population correlation matrix was computed:

$$\mathbf{R_R} = \mathbf{\Lambda \Phi \Lambda}',$$

where $\mathbf{R_R}$ is the reproduced population correlation matrix, $\mathbf{\Lambda}$ is the k (variables) $\times$ r (factors) factor loading matrix, and $\mathbf{\Phi}$ is the $r \times r$ correlation matrix. The population correlation matrix, $\mathbf{R_P}$, was then obtained by putting the unities on the diagonal of $\mathbf{R_R}$. Next, Cholesky decomposition was performed on the correlation matrix such that:

$$\mathbf{R_P} = \mathbf{U}'\mathbf{U}.$$

If the population correlation matrix was not positive definite (i.e., at least one eigenvalue ≤ 0) or any single item's communality was greater than 0.90, then $\mathbf{\Lambda}$ was re-generated and the same procedure was followed until these criteria are met. Finally, the sample data matrix of continuous variables was computed:

$$\mathbf{X} = \mathbf{ZU},$$

where $\mathbf{Z}$ is a matrix of random multivariate normal data with rows equal to the sample size and columns equal to the number of variables.

We manipulated number of variables per factor (4, 6, and 8), number of factors (2, 4, and 6), and correlations between factors (.00, .30, .50, and .70). As the magnitude of the correlations between factors increased, so too did the variance of the distribution the cross-loadings were drawn from. Specifically, cross-loadings were drawn from a random normal distribution with a mean of 0 and standard deviation of .050, .075, .100, and .125, respectively. This made it possible to generate cross-loading magnitudes that were quite large (e.g., .40), creating more difficult conditions to decipher factor from network models when the correlations between factors were large (e.g., .70). Cross-loadings were allowed to be both positive and negative. Factor loadings on the dominant factors were randomly drawn from a uniform distribution with a minimum of .40 and maximum of .70. In total, there were 72 conditions (sample size $\times$ number of factors $\times$ variables per factor $\times$ correlations between factors) that were generated 100 times.

4.1.2 Network model We generated data from two different network models: small-world and random. We generated small-world networks by adapting the `bdgraph.sim` algorithm in the *BDgraph* package (Mohammadi & Wit, 2015)

in R (R Core Team, 2020) to incorporate the `sample_smallworld` function from the *igraph* package (Csardi & Nepusz, 2006). The algorithm starts by generating a binary undirected small-world network that follows the Watts-Strogatz model (Watts & Strogatz, 1998). Next, following Williams, Rhemtulla, Wysocki, and Rast (2019), the weights are drawn from a G-Wishart distribution corresponding to 90% of partial correlations within the range ±.40. As Williams, Rhemtulla, Wysocki, and Rast (2019) note, large networks are more likely to have smaller partial correlations due to more variance being partialled out; however, given that many psychological assessment instruments have redundancies (Christensen, Golino, & Silvia, 2020), partial correlations as large as .40 may not be uncommon even when there are a large number of variables (Wysocki & Rhemtulla, 2019). Therefore, we allowed networks, regardless of the number of variables, to have weights between ±.40. The distributions of the absolute values of these weights were typically positively skewed.

For the small-world network models, number of variables (12, 24, 36, and 48), rewiring probabilities (.075, .15, and .30), and densities (.30, .50, and .70) were manipulated. The rewiring probabilities were chosen on the basis of typical small-world network models where the standard Watts-Strogatz small-world model is around .10 (±5) and typical psychological small-world networks are likely somewhere between .01 and .50. Importantly, the number of variables tended to be within the same range as the factor models (between 8–48) to allow for closer comparisons of the two models, which had a similar number of variables. It is worth noting that our density and partial correlation magnitudes were within the general range of many psychological networks (for a review, see Wysocki & Rhemtulla, 2019). In total, there were 72 conditions (sample size × number of variables × rewiring probabilities × densities) that were generated 100 times.

We generated random networks using the `bdgraph.sim` algorithm in the *BDgraph* package. The network and data generation approach was identical to the small-world networks. The main difference is that random networks randomly connect edges between all nodes, making them less structured relative to small-world networks (Watts & Strogatz, 1998). Like the small-world networks, we manipulated the number of variables (15, 25, 35, and 45) and density of the random networks (.30, .50, and .70). We also manipulated the probability that a pair of nodes would have edge (.25, .50, .75). In total, there were 72 conditions (sample size × number of variables × rewiring probabilities × densities) that were generated 100 times.

4.2 Statistical Analysis

4.2.1 Analysis of Variance We computed analysis of variances (ANOVAs) across conditions. We used a fully factorial design to allow for all possible interactions between conditions. Partial eta squared (η_p^2) was used for effect size. We followed Cohen's (1992) effect size guidelines: small ($\eta_p^2 = 0.01$), moderate($\eta_p^2 = 0.06$), and large ($\eta_p^2 = 0.14$).

4.2.2 Confusion Matrix Metrics We computed confusion matrix metrics for the models using the empirical, bootstrap, and proportion predictions of the algorithm. To provide an example of these metrics, we use the factor model as the model under consideration. A true positive (TP) was when the predicted and true generating model matched the model under consideration (e.g., factor). A true negative (TN) was when the predicted and true generating model (e.g., network) were not the model under consideration (e.g., factor). A false positive (FP) was when the predicted generating model matched the model under consideration (e.g., factor) but not the true generating model (e.g., network). A false negative (FN) was when the predicted generating model (e.g., network) did not match the model true generating model and model under consideration (e.g., factor).

Using this confusion matrix, we computed sensitivity ($\frac{TP}{TP+FN}$), specificity ($\frac{TN}{TN+FP}$), false discovery rate (FDR; $\frac{FP}{FP+TP}$), accuracy ($\frac{TP+TN}{TP+FP+TN+FN}$), and Matthews correlation coefficient (MCC; $\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}}$). Sensitivity is the proportion of positives that are correctly identified as TPs, while specificity is the proportion of negatives that are correctly identified as TNs. The FDR is the proportion of FPs that are found relative to the total positives that are predicted by the algorithm. Accuracy is the proportion of correct predictions (TPs and TNs) of the algorithm, representing an overall summary of sensitivity and specificity. Finally, the MCC is considered the best overall metric for classification evaluation because it is an unbiased measure that uses all aspects of the confusion matrix, representing a special case of the phi coefficient between the predicted and true model (Chicco & Jurman, 2020).

5 Results

Starting with general accuracy, the neural network predictions had the highest percent correct: proportion (96.2%), bootstrap (95.1%), and empirical (86.7%). These were followed by the original algorithm (85.9%) and the logistic regression predictions: bootstrap (70.5%), proportion (68.7%), and empirical (67.4%). Because the logistic regression predictions were poor, we focus on the confusion matrix metrics of the neural network and original algorithm predictions.

Across all metrics, the bootstrap and proportion predictions were superior to the single-shot empirical predictions. It is important to note that accuracy and MCC will be the same between models, specificity and sensitivity will be the opposite between models, and FDR will be different between the two models. For specificity and sensitivity, we focus on factor models (sensitivity and specificity for network models, respectively). Overall, proportion predictions outperformed all others: sensitivity = 0.995 and specificity = 0.946 for factor models. The accuracy and MCC were also very high: 0.962 and 0.919, respectively. The FDR was 0.099 for factor models and 0.003 for network models.

The bootstrap predictions performed similarly well: sensitivity = 0.987 and specificity = 0.933 for factor models. The accuracy and MCC were high: 0.951

and 0.896, respectively. The FDR was 0.120 for factor models and 0.007 for network models. The empirical predictions were slightly better than the original algorithm (in parentheses): sensitivity = 0.984 (0.931) and specificity = 0.809 (0.822) for factor models. The accuracy and MCC were fairly high: 0.867 (0.859) and 0.751 (0.718), respectively. The FDR was 0.279 (0.273) for factor models and 0.010 (0.041) for network models.

5.1 Factor Model Percent Correct

In general, predictions for the factor model were highly accurate ($\geq 75\%$) across all conditions for the neural network and logistic regression methods (Figure 1). Lower accuracy for all methods tended to occur when correlations between factors were large (.70). The ANOVA found that there was only one effect that reached at least a moderate effect size. This moderate effect was an interaction between method and correlations between factors ($\eta_p^2 = 0.07$). This interaction was driven by the original algorithm and large correlations between factors (Figure 1).

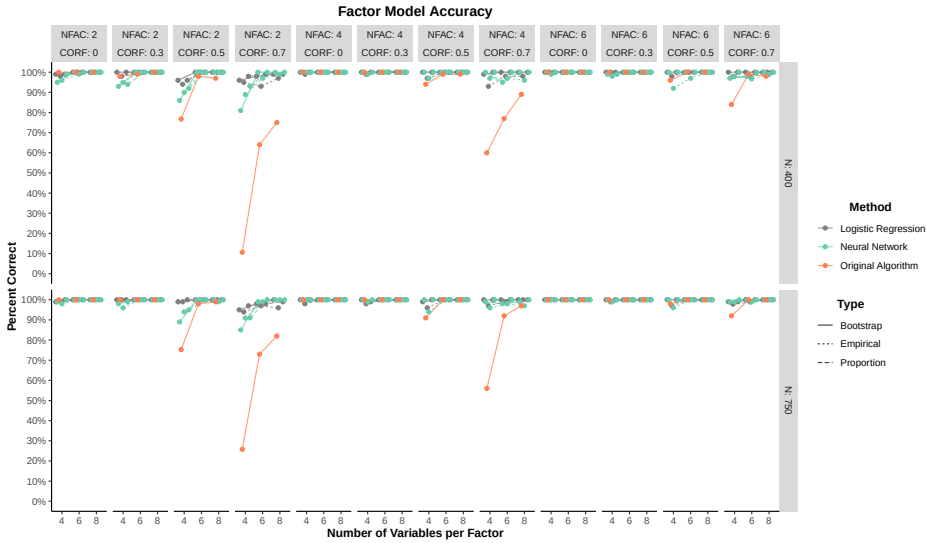


Figure 1. Percent correct for factor models in each condition. NFAC = number of factors, CORF = correlations between factors, and N = sample size.

Across all conditions, the neural network and logistic regression methods were comparable to or better than the original LCT algorithm. The neural network method was comparable to logistic regression method on all three prediction types: empirical (98.4% and 98.9%, respectively), bootstrap (98.7% and 99.6%, respectively), and proportion (99.5% and 99.8%). The original algorithm was lower but still had high accuracy (93.1%).

5.2 Small-world Network Model Percent Correct

As a general trend, all methods tended to improve in percent correct as the small-world network models became denser (Figure 2). The neural network method by far outperformed the logistic regression and original algorithm methods when the networks were sparse (0.30). Across all conditions, the neural networks performed as well as or better than the logistic regression and original algorithm predictions, with the proportion predictions achieving at least 75% correct or greater. There was one large effect for method ($\eta_p^2 = 0.18$). The overall percent correct made this effect clear: neural network (90.9%), logistic regression (82.1%), and original algorithm (56.5%).

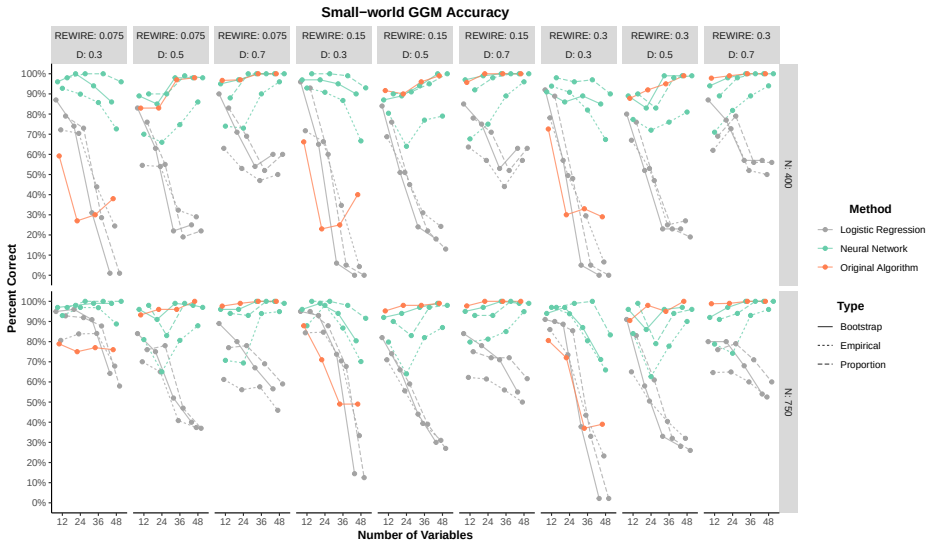


Figure 2. Percent correct for small-world network models in each condition. REWIRE = rewiring probability, D = density, and N = sample size.

Relative to the neural network method, the logistic regression and original algorithm methods did not perform as well. These results suggest that the logistic regression and original algorithm were strongly biased toward factor models. There were two clear patterns in their results. Logistic regression performed worse as the density decreased and the number of variables increased. The original algorithm was primarily affected by density with accuracy decreasing as density decreased.

5.3 Random Network Model Percent Correct

The random network models were not used to train or develop the methods and therefore represent the strongest test of generalizability. As a general trend, all

methods tended to improve in percent correct as the random network models became denser (Figure 3). Overall, the neural network (88.3%) outperformed the original algorithm (82.3%) and logistic regression (50.8%) methods. When broken down by prediction type, neural network proportion (93.4%) and bootstrap (91.8%) predictions had the highest accuracy followed by the original algorithm (82.3%) and neural network empirical (79.6%) prediction. All logistic regression predictions were less than 60%. There was one large effect for method ($\eta_p^2 = 0.20$). This effect was largely driven by logistic regression (Figure 3).

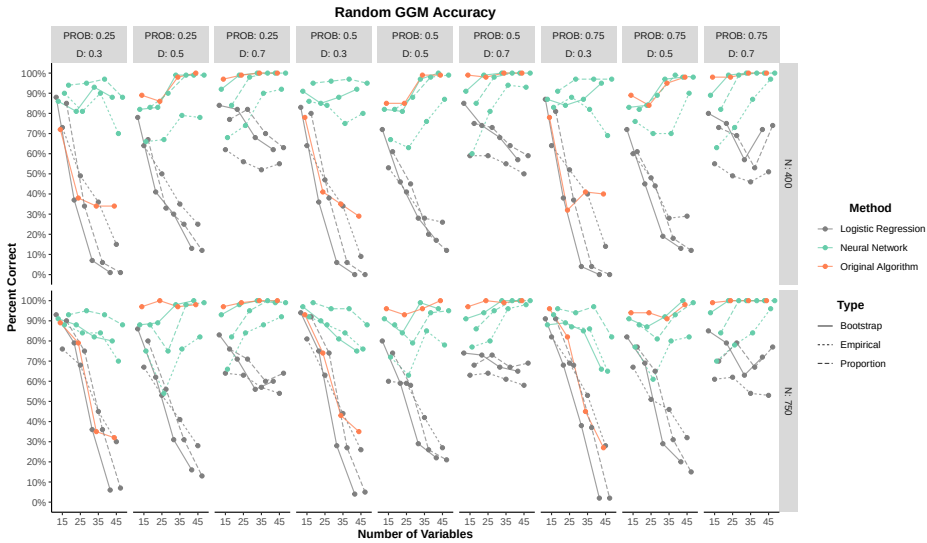


Figure 3. Percent correct for small-world network models in each condition. PROB = edge probability, D = density, and N = sample size.

These results add further support to the finding that logistic regression was strongly biased toward factor models. Similar to the small-world network results, all methods tended to decrease in accuracy as the density decreased. Accuracy tended to increase as variables increased for the neural network while accuracy tended to decrease as variables increased for logistic regression.

6 Real-world Examples

The simulation provides evidence that the LCT algorithm paired with neural networks can be a powerful predictive tool for identifying whether data are generated from a specific model. It is important, however, to demonstrate that the LCT works in practice. To illustrate this, we examined two different datasets that are assumed to be generated from a factor and network model.

6.1 International Personality Item Pool Big Five Inventory

The first example dataset consisted of 2800 observations on items from the International Personality Item Pool's (Goldberg, 1999) Big Five Inventory (BFI; John, Donahue, & Kentle, 1991), which is available in the *psych* package (Revelle, 2017) in R. The BFI traditionally has five factors, each with five items, corresponding to the Big Five factor model: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism. The robustness of this factor structure has been demonstrated across a variety of samples (e.g., Donnellan, Oswald, Baird, & Lucas, 2006). Although there is no way to determine that the BFI is actually generated from a factor model, its robust factor structure suggests that the data structure should follow a factor model.

We applied the LCT to the full dataset as well as sub-samples that were randomly split into 400 cases each (seven sub-samples in total; see Appendix A3 for code to replicate this analysis). For the full dataset, all predictions—empirical, bootstrap, and proportion—were for a factor model. Across the sub-samples, the results varied slightly by prediction: empirical (6 factor and 1 network), bootstrap (7 factor), and proportion (7 factor).

6.2 Resting State Default Mode Network

The second example dataset consisted of mean blood oxygen level-dependent (BOLD) activation levels of twenty regions of interest (ROIs) in the brain that corresponded to the default mode network (DMN) during five-minute resting state scans in 144 participants from Beaty et al. (2018). The DMN corresponds to a set of cortical midline, medial temporal, and posterior inferior parietal regions that often co-activate together. Recent research has demonstrated that the DMN can be broken down into several distinct sub-networks (Andrews-Hanna, Smallwood, & Spreng, 2014; Gordon et al., 2020). Brain networks are a well-known real-world example of networks, which make them an appropriate test of whether the LCT performs as expected.

We applied the LCT to the correlation matrices of the 20 ROIs based on the DMN structure identified in the Shen brain atlas (Shen, Tokoglu, Papademetris, & Constable, 2013; see Appendix A.4 for code to replicate this analysis). The correlation matrices were derived from time series with the length of 150, which was used as the sample size input for the LCT. For the bootstrap and proportion predictions, all participants' DMN networks were suggested to be generated from a network model. The empirical prediction suggested that most 140 (97.2%) were generated from network models.

6.3 Summary

Taken together, these examples demonstrate the validity of the LCT on real-world datasets that were expected to be generated from factor and network models. Given the robustness of the proportion prediction of the LCT in the simulation and our examples here, we suggest that researchers should place the

most weight on this prediction. A consensus across predictions is most likely to be robust but when they conflict researchers should give priority to the proportion prediction followed by the bootstrap and empirical predictions. One benefit of the proportion prediction is that it provides some inference into the certainty of the data-generating model by offering the proportion of samples that were predicted to be from either a factor or network model.

7 Discussion

The present study sought to use artificial neural networks to improve the LCT algorithm, which was designed to determine whether data are generated from a factor or network model based on factor and network loading structures. Our results demonstrate how artificial neural networks can be a powerful tool for developing highly predictive models. In the context of our study, we demonstrated that neural networks (specifically with proportion predictions) outperform simple heuristics (i.e., the original LCT algorithm) and logistic regression models for predicting the data-generating model.

The significance of this problem has grown increasingly relevant as recent studies have demonstrated that similar covariance patterns and models parameters (e.g., dimensions, loadings) can be derived from factor and network models (Golino et al., 2020; Hallquist, Wright, & Molenaar, 2019; Marsman et al., 2018; van Bork et al., 2019; Waldorp & Marsman, 2020). These findings have shifted the focus of the differences between these models from statistical to theoretical (Guyon, Falissard, & Kop, 2017; Kruis & Maris, 2016). Indeed, when the data generating mechanism is a factor model, then the model parameters of factor and network models can be shown to be consistent with one another (e.g., dimensions, loadings; Christensen & Golino, 2021; Golino et al., 2020). These parameters, specifically loadings, start to differ when the data generating mechanism is not a factor model. This raises an important question: What is the difference between the structure of factor and network models?

We pinned our rationale on the factor model's focus on extracting common covariance. When it comes to our neural networks, their interpretations are a black box of linear and non-linear transformations of the input to the output and therefore make our predictions accurate but not necessarily explanatory (but see Buhrmester, Münch, & Arens, 2019; Yarkoni & Westfall, 2017). Although some hints are provided by our feature importance analysis (see Appendix A.2), the exact mapping of between the loading structures and predicted model is likely multifaceted (as demonstrated by the better performance in training and validation of the neural networks over logistic regression). In unidimensional models, there appears to be some statistical differences that can be exploited but this may not generalize to more complex models (van Bork et al., 2019). We show that, at the very, least summaries of the data's structure (proportions of small, dominant, and cross-loadings) are important for differentiating data generated from these models.

When considering statistical assumptions and the feature importance analysis, our results point to the cross-loadings between dimensions: factor models tend to minimize cross-loadings whereas network models typically have many (Christensen & Golino, 2021). Indeed, cross-loadings of the factor models were either the first or second most important input for the neural networks predicting whether the data were generated from a factor or network model (see Appendix A.2). Another difference is the extent to which there is clustering due to common covariance: factor models attempt to specifically extract common covariance whereas network models partition covariance. This is made evident by the importance of the dominant factor loading across the models. This strongly suggests that the lack of common covariance in dimensions of network models is a substantial contributor for differentiating them from factor models. This finding is consistent with variables in network models being characterized as "causally autonomous" (Cramer et al., 2012).

Although our findings may not be able to provide an exact statistical answer about the differences between these models (e.g., van Bork et al., 2019), they do provide a predictive tool for whether data are structured as a factor or network model. Specifically, the proportion predictions of the neural network following the LCT algorithm had high accuracy for all models. Importantly, we do not suggest that the LCT can inform the researcher about whether their data was *actually* generated from a specific model. This is a critical distinction: The LCT can accurately predict whether the data are *structured* as a specific model rather than actually being generated by it. Indeed, our simulated data were generated from specific models but this does not mean that data structured like a factor model could not be generated from a network model (and vice versa; Fried, 2020; van Bork et al., 2019; van der Maas et al., 2006).

This issue of equally plausible data-generating mechanisms has been discussed at length in the literature (Christensen & Golino, 2021; Marsman et al., 2018; van Bork et al., 2019; Waldorp & Marsman, 2020), leading to recent calls for researchers to develop formal (i.e., computational and mathematical) theories about their psychological phenomena of interest (Borsboom, van der Maas, Dalege, Kievit, & Haig, 2020; Fried, 2020; Haslbeck, Ryan, Robinaugh, Waldorp, & Borsboom, 2019). Theories and hypotheses about the relations between components of the phenomena should be developed a priori to test their relations. These should then inform whether a factor or network model is a more appropriate statistical model for the representation of those relations. We view the LCT as a test for whether components are structured like a factor or network model, which can inform the researcher as to whether the relations between components are interacting as expected. Said differently, we do not advise that the LCT supplant theory about the relations between variables but suggest that it can serve as a tool for reasoning about the hypothesized structure of psychological measurements.

In this respect, scale developers can structure their scales to align more with the structure of a factor or network model—that is, the data structure can be manipulated to produce data that appear to be generated from one model or

the other (see Appendix A.6 for an example). In fact, contemporary psychometric practice has been doing exactly this for many years: variables that are strongly interrelated are usually retained in scales and variables with substantial cross-loadings are usually removed from scales (DeVellis, 2017). This approach is often justified to ensure that the phenomena of interest are being cleanly measured (i.e., unidimensional) yet most researchers rarely discuss whether the theory about the relations between the variables actually dictate such distinctions. Therefore, it again comes down to theory as to whether the data are actually generated from said model.

For more practical terms, researchers must consider the data-generating model when estimating scores from these psychometric models (network scores can be computed as a weighted composite; e.g., Golino, Christensen, Moulder, Kim, & Boker, 2020). As shown in Appendix A.2 and Christensen and Golino (2021), the loading structures for factor and network loadings are consistent with one another when the data are generated from a factor model, which suggests that there is little consequence in whether a factor or network model is used to estimate scores (Golino, Christensen, Moulder, Kim, & Boker, 2020). When the data are generated (or even structured) as a network model, then there is divergence between the loading structures with variables (e.g., dominant loadings; Appendix A.2). This divergence can have a substantial effect on the computation and interpretation of scores.

Such a consequence has been noted in less drastic circumstances with sum scores and factor scores where differences can be observed when a tau-equivalent latent variable model (i.e., sum scores) is applied to data generated from a congeneric latent variable model (i.e., factor scores; McNeish & Wolf, 2020). These differences in factor structures can potentially have substantial consequences for the reliability and validity of measurement. Moreover, these consequences further underscore the importance for researchers to consider that "scoring scales—by any method—is a statistical procedure that requires evidence and justification" (McNeish & Wolf, 2020, p. 2). Therefore, if data are generated from a network model, then factor scores may not be appropriate and could possibly jeopardize the validity of the research. Our study demonstrates that the LCT can be used as one method to provide such evidence and justification as well as guide researchers toward more valid measurement.

Importantly, we also echo recent calls by researchers who have stated that there is no need to pit these models against each other but rather develop hybrid models that include components that are from common cause and causal systems (Christensen, Golino, & Silvia, 2020; Epskamp, Rhemtulla, & Borsboom, 2017; Fried, 2020; Guyon, Falissard, & Kop, 2017). In this way, researchers should consider the level of organization at which each phenomena is being measured. Factor models, for example, may be more appropriate when measuring a specific phenomenon with highly similar variables like a single characteristic of personality whereas network models may be more appropriate for understanding how these specific characteristics coalesce into more complex systems like a personality trait (Christensen, Golino, & Silvia, 2020; Mõttus & Allerhand, 2017). Even

still, individual personality traits may then appear as a factor model when examined together. This suggests that the level of organization may influence the data structure and the relationships between the psychological components. This jibes with the notion that hybrid models may be the most optimal stance (Fried, 2020; Guyon, Falissard, & Kop, 2017). The LCT can help researchers explore and verify these hypothesized structures to better determine how hybridization should occur.

There are several limitations that researchers must consider when using the LCT. First, the LCT was trained on small-world network models and therefore carries the assumption that most psychological networks will be generated from small-world network models. We think this assumption is reasonable because many real-world networks show small-world structure (e.g., brain networks; Muldoon, Bridgeford, & Bassett, 2016) and many psychological phenomena exhibit properties that align with these assumptions such as psychopathological disorders (Borsboom, Cramer, Schmittmann, Epskamp, & Waldorp, 2011): clustering of symptoms within a disorder (high clustering coefficient) yet bridges between symptoms to other disorders (low average shortest path lengths; Cramer, Waldrop, van der Maas, & Borsboom, 2010). Moreover, we demonstrate that the LCT can generalize to random network structures, which may be more appropriate when the network consists of unique variables that represent a specific dimension like a network comprised of individual latent variables that represent causally distinct phenomena (Cramer et al., 2012).

There are few standards for the characteristics and topology of what can be considered a "typical" psychological network. Our data generating assumptions were based on previous evidence that most real-world networks tend to be small-world (including psychological networks; Borsboom, Cramer, Schmittmann, Epskamp, & Waldorp, 2011), but the extent to which psychological networks are represented by small-world networks and whether the parameters used in the study mimic real-world psychological networks requires empirical validation (but see Wysocki & Rhemtulla, 2019). In large part, this is because few psychological network studies have examined the topological features of psychological networks such as their degree distribution, which is a critical characteristic for determining the type of network (e.g., random, small-world, scale-free, exponential random graph; Newman, 2010). Further, small-worldness measures should be used to determine whether data are more like a random, lattice, or small-world network (see Telesford, Joyce, Hayasaka, Burdette, & Laurienti, 2011). In practice, this task is difficult because network estimation methods differ in their preference for sparsity, which affects all network measures (Wysocki & Rhemtulla, 2019). Better data generation follows from more studies examining and reporting the topology of psychological networks (e.g., Battiston et al., 2020; Burger et al., 2020), which can in turn be used to train better neural networks to make more valid predictions.

This leads us to a second, influential limitation: the predictions of the neural networks are only as good as the data they are trained on. Therefore, we must be critical of our own data generating methods and question whether they resemble

real-world data. We believe that we have provided reasonably realistic datasets that include factor models with dominant loadings between .40 and .70 and a varying degree of cross-loadings. The range of loadings represent what are considered to be acceptable to very high (Comrey & Lee, 2013), with .40 being considered a rule of thumb for appropriate measurement of a latent variable (DeVellis, 2017). Still, not all datasets will have loadings on the dominant factor that are within this range.

Finally, in light of our discussion on theory, the LCT is focused on cross-sectional datasets when most phenomena are likely to be dynamical systems (e.g., Haslbeck, Ryan, Robinaugh, Waldorp, & Borsboom, 2019). This is a limitation of the current implementation of the LCT but we suspect that the LCT can be generalized to time series data by using dynamic factor analysis and dynamic EGA (Golino, Christensen, Moulder, Kim, & Boker, 2020). Such an approach could lead to determining whether some people represent represent a phenomenon of interest as a common cause or causal system. This in turn could offer inferences into individualized psychopathological intervention (Wright & Woods, 2020), providing more specific answers to whether it would be more effective for a clinician to treat an underlying disorder or specific symptoms.

Author Note

All data, code, and materials can be found on the Open Science Framework: `https://osf.io/4fe9g/`.

The authors made the following contributions. Alexander P. Christensen (`https://orcid.org/0000-0002-9798-7037`): Conceptualization, Software, Methdology, Writing - Original Draft, Writing - Review & Editing; Hudson Golino (`https://orcid.org/0000-0002-1601-1447`): Conceptualization, Software, Methdology, Writing - Review & Editing.

Correspondence concerning this article should be addressed to Alexander P. Christensen, Department of Neurology, University of Pennsylvania, Philadelphia, PA, 19104. E-mail: `alexpaulchristensen@gmail.com`

References

Allaire, J. J., & Chollet, F. (2020). *keras: R interface to 'Keras'*. Retrieved from `https://keras.rstudio.com`

Andrews-Hanna, J. R., Smallwood, J., & Spreng, R. N. (2014). The default network and self-generated thought: Component processes, dynamic control, and clinical relevance. *Annals of the New York Academy of Sciences*, *1316*, 29–52. doi: https://doi.org/10.1111/nyas.12360

Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., ... Petri, G. (2020). Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*. doi: https://doi.org/10.1016/j.physrep.2020.05.004

Beaty, R. E., Kenett, Y. N., Christensen, A. P., Rosenberg, M. D., Benedek, M., Chen, Q., ... Silvia, P. J. (2018). Robust prediction of individual creative ability from brain functional connectivity. *Proceedings of the National Academy of Sciences*, *115*, 1087–1092. doi: https://doi.org/10.1073/pnas.1713532115

Borsboom, D. (2006). The attack of the psychometricians. *Psychometrika*, *71*, 425–440. doi: https://doi.org/10.1007/s11336-006-1447-6

Borsboom, D. (2008). Psychometric perspectives on diagnostic systems. *Journal of Clinical Psychology*, *64*, 1089–1108. doi: https://doi.org/10.1002/jclp.20503

Borsboom, D. (2017). A network theory of mental disorders. *World Psychiatry*, *16*, 5–13. doi: https://doi.org/10.1002/wps.20375

Borsboom, D., Cramer, A. O. J., Schmittmann, V. D., Epskamp, S., & Waldorp, L. J. (2011). The small world of psychopathology. *PLoS ONE*, *6*, e27407. doi: https://doi.org/10.1371/journal.pone.0027407

Borsboom, D., van der Maas, H., Dalege, J., Kievit, R., & Haig, B. (2020). Theory construction methodology: A practical framework for theory formation in psychology. *PsyArXiv*. doi: https://doi.org/10.31234/osf.io/w5tp8

Buhrmester, V., Münch, D., & Arens, M. (2019). Analysis of explainers of black box deep neural networks for computer vision: A survey. *arXiv*. Retrieved from `https://arxiv.org/abs/1911.12116`

Burger, J., Isvoranu, A.-M., Lunansky, G., Haslbeck, J., Epskamp, S., Hoekstra, R., ... Blanken, T. (2020). Reporting standards for psychological network analyses in cross-sectional data. *PsyArXiv*. doi: https://doi.org/10.31234/osf.io/4y9nz

Chen, J., & Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, *95*, 759–771. doi: https://doi.org/10.1093/biomet/asn034

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*, 6. doi: https://doi.org/10.1186/s12864-019-6413-7

Christensen, A. P., Garrido, L. E., & Golino, H. (2021). Comparing community detection algorithms in psychological data: A Monte Carlo simulation. *PsyArXiv*. doi: https://doi.org/10.31234/osf.io/hz89e

Christensen, A. P., & Golino, H. (2019). Estimating the stability of the number of factors via Bootstrap Exploratory Graph Analysis: A tutorial. *PsyArXiv*. doi: https://doi.org/10.31234/osf.io/9deay

Christensen, A. P., & Golino, H. (2021). On the equivalency of factor and network loadings. *Behavior Research Methods*. doi: https://doi.org/10.3758/s13428-020-01500-6

Christensen, A. P., Golino, H., & Silvia, P. J. (2020). A psychometric network perspective on the validity and validation of personality trait questionnaires. *European Journal of Personality*, *34*, 1095–1108. doi: https://doi.org/10.1002/per.2265

Cohen, J. (1992). A power primer. *Psychological Bulletin*, *112*, 155–159. doi: https://doi.org/10.1037/0033-2909.112.1.155

Comrey, A. L., & Lee, H. B. (2013). *A first course in factor analysis* (2nd ed.). New York, NY: Psychology Press.

Cramer, A. O. J., van der Sluis, S., Noordhof, A., Wichers, M., Geschwind, N., Aggen, S. H., ... Borsboom, D. (2012). Dimensions of normal personality as networks in search of equilibrium: You can't like parties if you don't like people. *European Journal of Personality*, *26*, 414–431. doi: https://doi.org/10.1002/per.1866

Cramer, A. O. J., Waldrop, L. J., van der Maas, H. L., & Borsboom, D. (2010). Comorbidity: A network perspective. *Behavioral and Brain Sciences*, *33*, 137–150. doi: https://doi.org/10.1017/S0140525X09991567

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, *1695*, 1–9. Retrieved from `https://www.semanticscholar.org/paper/The-igraph-software-package-for-complex-network-Cs/%C3/%A1rdi-Nepusz/1d2744b83519657f5f2610698a8ddd177ced4f5c?p2df`

DeVellis, R. F. (2017). *Scale development: Theory and applications* (4th ed.). Thousand Oaks, CA: SAGE Publications.

Donnellan, M. B., Oswald, F. L., Baird, B. M., & Lucas, R. E. (2006). The mini-IPIP scales: Tiny-yet-effective measures of the Big Five factors of personality. *Psychological Assessment*, *18*, 192–203. doi: https://doi.org/10.1037/1040-3590.18.2.192

Dozat, T. (2016). Incorporating Nesterov momentum in Adam. In *Proceedings of 4th international conference on learning representations, workshop track* (pp. 604–612). San Juan, Puerto Rico. Retrieved from `https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ`

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics*, *35*, 352–359. doi: https://doi.org/10.1016/S1532-0464(03)00034-0

Edwards, J. R., & Bagozzi, R. P. (2000). On the nature and direction of relationships between constructs and measures. *Psychological Methods*, *5*, 155–174. doi: https://doi.org/10.1037/1082-989X.5.2.155

Epskamp, S., & Fried, E. I. (2018). A tutorial on regularized partial correlation networks. *Psychological Methods*, *23*, 617–634. doi: https://doi.org/10.1037/met0000167

Epskamp, S., Rhemtulla, M., & Borsboom, D. (2017). Generalized network psychometrics: Combining network and latent variable models. *Psychometrika*, *82*, 904–927. doi: https://doi.org/10.1007/s11336-017-9557-x

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*, 1–81. Retrieved from `https://jmlr.org/papers/v20/18-760.html`

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, *486*, 75–174. doi: https://doi.org/10.1016/j.physrep.2009.11.002

Fried, E. I. (2020). Lack of theory building and testing impedes progress in the factor and network literature. *PsyArXiv*. doi: https://doi.org/10.31234/osf.io/zg84s

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*, 432–441. doi: https://doi.org/10.1093/biostatistics/kxm045

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularized paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *22*, 1–34. doi: https://doi.org/10.18637/jss.v033.i01

Friedman, J., Hastie, T., & Tibshirani, R. (2014). *glasso: Graphical lasso – estimation of Gaussian graphical models*. Retrieved from `https://CRAN.R-project.org/package=glasso`

Goldberg, L. R. (1999). A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. In I. Mervielde, I. Deary, F. De Fruyt, & F. Ostendorf (Eds.), *Personality psychology in Europe* (Vol. 7, pp. 7–28). Tilburg, The Netherlands: Tilburg University Press.

Golino, H., Christensen, A. P., Moulder, R., Kim, S., & Boker, S. M. (2020). Modeling latent topics in social media using Dynamic Exploratory Graph Analysis: The case of the right-wing and left-wing trolls in the 2016 US elections. *PsyArXiv*. doi: https://doi.org/10.31234/osf.io/tfs7c

Golino, H., & Epskamp, S. (2017). Exploratory Graph Analysis: A new approach for estimating the number of dimensions in psychological research. *PLoS ONE*, *12*, e0174035. doi: https://doi.org/10.1371/journal.pone.0174035

Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., ... Martinez-Molina, A. (2020). Investigating the performance of Exploratory Graph Analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, *25*, 292–320. doi: https://doi.org/10.1037/met0000255

Gordon, E. M., Laumann, T. O., Marek, S., Raut, R. V., Gratton, C., Newbold, D. J., ... others. (2020). Default-mode network streams for coupling to language and control systems. *Proceedings of the National Academy of Sciences*, *117*, 17308–17319. doi: https://doi.org/10.1073/pnas.2005238117

Guyon, H., Falissard, B., & Kop, J.-L. (2017). Modeling psychological attributes in psychology–an epistemological discussion: Network analysis vs. latent variables. *Frontiers in Psychology*, *8*, 798. doi: https://doi.org/10.3389/fpsyg.2017.00798

Hallquist, M., Wright, A. C. G., & Molenaar, P. C. M. (2019). Problems with centrality measures in psychopathology symptom networks: Why network psychometrics cannot escape psychometric theory. *Multivariate Behavioral Research*. doi: https://doi.org/10.1080/00273171.2019.1640103

Haslbeck, J., Ryan, O., Robinaugh, D., Waldorp, L., & Borsboom, D. (2019). Modeling psychopathology: From data models to formal theories. *PsyArXiv*. doi: https://doi.org/10.31234/osf.io/jgm7f

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034).

Heaton, J. (2008). *Introduction to neural networks with Java* (2nd ed.). St. Louis, MO: Heaton Research, Inc.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*, 251–257. doi: https://doi.org/10.1016/0893-6080(91)90009-T

Hurley, R. S., Losh, M., Parlier, M., Reznick, J. S., & Piven, J. (2007). The broad autism phenotype questionnaire. *Journal of Autism and Developmental Disorders*, *37*(9), 1679–1690.

Ingersoll, B., Hopwood, C. J., Wainer, A., & Donnellan, M. B. (2011). A comparison of three self-report measures of the broader autism phenotype in a non-clinical sample. *Journal of Autism and Developmental Disorders*, *41*, 1646–1657. doi: https://doi.org/10.1007/s10803-011-1192-2

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. New York, NY: Springer. doi: https://doi.org/10.1007/978-1-4614-7138-7

Jessen, L. E. (2021). *nnvizRt: A server for visualizing architectures of neural networks*. Retrieved from `https://leonjessen.shinyapps.io/nnvizRt/`

John, O. P., Donahue, E. M., & Kentle, R. L. (1991). *The Big Five Inventory—Versions 4a and 54*. Berkeley, CA: University of California, Berkeley, Institute of Personality and Social Research.

Karpathy, A. (2019). A recipe for training neural networks. Retrieved April 25, 2019, from `https://karpathy.github.io/2019/04/25/recipe/`

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv*. Retrieved from `https://arxiv.org/abs/1412.6980`

Kruis, J., & Maris, G. (2016). Three representations of the Ising model. *Scientific Reports*, *6*, srep34175. doi: https://doi.org/10.1038/srep34175

Lauritzen, S. L. (1996). *Graphical models*. Oxford, UK: Clarendon Press.

Marsman, M., Borsboom, D., Kruis, J., Epskamp, S., van Bork, R., Waldorp, L. J., ... Maris, G. (2018). An introduction to network psychometrics: Relating Ising network models to item response theory models. *Multivariate Behavioral Research*, *53*, 15–35. doi: https://doi.org/10.1080/00273171.2017.1379379

McNeish, D., & Wolf, M. G. (2020). Thinking twice about sum scores. *Behavior Research Methods*. doi: https://doi.org/10.3758/s13428-020-01398-0

Mohammadi, R., & Wit, E. C. (2015). BDgraph: An R package for Bayesian structure learning in graphical models. *Journal of Statistical Software*, 1–30. doi: https://doi.org/10.18637/jss.v089.i03

Mõttus, R., & Allerhand, M. (2017). Why do traits come together? The underlying trait and network approaches. In V. Ziegler-Hill & T. K. Shackelford

(Eds.), *SAGE handbook of personality and individual differences: The science of personality and individual differences* (pp. 1–22). London, UK: SAGE Publications.

Muldoon, S. F., Bridgeford, E. W., & Bassett, D. S. (2016). Small-world propensity and weighted brain networks. *Scientific Reports*, *6*, 22057. doi: https://doi.org/10.1038/srep22057

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *International conference on machine learning* (pp. 807–814). Haifa, Israel. Retrieved from `https://icml.cc/Conferences/2010/papers/432.pdf`

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, *103*, 8577–8582. doi: https://doi.org/10.1073/pnas.0601602103

Newman, M. E. J. (2010). *Networks: An introduction*. New York, NY: Oxford University Press. doi: https://doi.org/10.1093/acprof:oso/9780199206650.001.0001

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv*. `https://arxiv.org/abs/1811.03378`

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, *10*, 191–218. doi: https://doi.org/10.7155/jgaa.00185

Prechelt, L. (2012). Early stopping – but when? In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade* (2nd ed., pp. 53–68). Berlin, Germany: Springer-Verlan.

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from `https://www.R-project.org/`

Revelle, W. (2017). *psych: Procedures for psychological, psychometric, and personality research*. Evanston, Illinois: Northwestern University. Retrieved from `https://CRAN.R-project.org/package=psych`

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv*. Retrieved from `https://arxiv.org/abs/1609.04747`

Shen, X., Tokoglu, F., Papademetris, X., & Constable, R. T. (2013). Groupwise whole-brain parcellation from resting-state fMRI data for network node identification. *NeuroImage*, *82*, 403–415. doi: https://doi.org/10.1016/j.neuroimage.2013.05.081

Sontag, E. D. (1991). Feedback stabilization using two-hidden-layer nets. In *1991 American control conference* (pp. 815–820). Boston, MA: IEEE. doi: https://doi.org/10.23919/ACC.1991.4791486

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139–1147). Atlanta, GA. Retrieved from `https://www.jmlr.org/proceedings/papers/v28/sutskever13.pdf`

Telesford, Q. K., Joyce, K. E., Hayasaka, S., Burdette, J. H., & Laurienti, P. J. (2011). The ubiquity of small-world networks. *Brain Connectivity*, *1*(5), 367–375. doi: https://doi.org/10.1089/brain.2011.0038

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288. doi: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

van Bork, R., Rhemtulla, M., Waldorp, L. J., Kruis, J., Rezvanifar, S., & Borsboom, D. (2019). Latent variable models and networks: Statistical equivalence and testability. *Multivariate Behavioral Research*, 1–24. doi: https://doi.org/10.1080/00273171.2019.1672515

van der Maas, H. L. J., Dolan, C. V., Grasman, R. P. P. P., Wicherts, J. M., Huizenga, H. M., & Raijmakers, M. E. J. (2006). A dynamical model of general intelligence: The positive manifold of intelligence by mutualism. *Psychological Review*, *113*, 842–861. doi: https://doi.org/10.1037/0033-295X.113.4.842

Waldorp, L., & Marsman, M. (2020). Relations between networks, regression, partial correlation, and latent variable model. *arXiv*. Retrieved from `https://arxiv.org/abs/2007.10656`

Ward, J. H. (1963). Hierarchical clustering to optimise an objective function. *Journal of the American Statistical Association*, *58*, 238–244.

Watt, J., Borhani, R., & Katsaggelos, A. (2016). *Machine learning refined: Foundations, algorithms, and applications*. Cambridge, UK: Cambridge University Press. doi: https://doi.org/10.1017/CBO9781316402276

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*, 440–442. doi: https://doi.org/10.1038/30918

Williams, D. R., Rhemtulla, M., Wysocki, A. C., & Rast, P. (2019). On nonregularized estimation of psychological networks. *Multivariate Behavioral Research*, *54*, 719–750. doi: https://doi.org/10.1080/00273171.2019.1575716

Wright, A. G., & Woods, W. C. (2020). Personalized models of psychopathology. *Annual Review of Clinical Psychology*, *16*. doi: https://doi.org/10.1146/annurev-clinpsy-102419-125032

Wysocki, A. C., & Rhemtulla, M. (2019). On penalty parameter selection for estimating network models. *Multivariate Behavioral Research*, 1–15. doi: https://doi.org/10.1080/00273171.2019.1672516

Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, *12*, 1100–1122. doi: https://doi.org/10.1177/1745691617693393

Zhou, Z.-H., Wu, J., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, *137*, 239–263. doi: https://doi.org/10.1016/S0004-3702(02)00190-X

A Appendix

A.1 Exploratory Graph Analysis

Exploratory graph analysis (EGA; Golino & Epskamp, 2017; Golino et al., 2020) is a network psychometrics dimension identification method. The approach be-

gins by estimating a network from the empirical data and applying a community detection algorithm to identify *communities* (or dimensions) in the network. The traditional EGA method estimates a Gaussian graphical model (GGM; Lauritzen, 1996) where nodes are variables and edges are the partial correlations between nodes after being conditioned on all other nodes. In psychological networks, the most common way of estimating a GGM is to use the graphical least absolute shrinkage and selection operator (GLASSO; Friedman, Hastie, & Tibshirani, 2008; Friedman, Hastie, & Tibshirani, 2014) with extended Bayesian information criterion (EBICglasso; Chen & Chen, 2008; Epskamp & Fried, 2018). Once the EBICglasso is applied, the Walktrap (Pons & Latapy, 2006) community detection algorithm is applied. The Walktrap algorithm uses random walks or stochastic steps from one node over an edge to another to determine the distance and similarity between two nodes. These random walks tend to stay within subsets of related nodes because they tend to be closer and more similar to one another. The algorithm merges the results, based on an agglomerative clustering approach (Ward, 1963), of the random walks to separate the communities. *Modularity* or the extent to which nodes maximize the proportion of connections within their community relative to connections to other communities (Newman, 2006) is then used as criterion for selecting the optimal clustering (or community) organization.

A.2 Training the Neural Networks

A.2.1 Data Generation Following the same data generating procedures in the main text, we generated 480,000 datasets in total. For the factor models, we manipulated number of variables per factor (3, 4, 5, 6, and 7), number of factors (3, 4, 5, and 6), and correlations between factors (.00, .30, .50, and .70). In total, there were 240 conditions (sample size $\times$ number of variables per factor $\times$ number of factors $\times$ correlations between factors). For each condition, 1,000 samples were generated.

In contrast to previous simulation studies on psychological networks which have generated data from random network models (e.g., Epskamp, Rhemtulla, & Borsboom, 2017; van Bork et al., 2019; Williams, Rhemtulla, Wysocki, & Rast, 2019), we generated the training network models based on small-world networks. Despite being the most widely studied type of network, random network models are largely incongruous with most real-world networks (e.g., lack of clustering, no correlation between degrees of adjacent nodes, shape of degree distribution; Newman, 2010). Small-world networks, however, at least mirror some properties of real-world networks (e.g., clustering, shortcuts between nodes; Newman, 2010) and are commonly found in real-world networks (e.g., brain networks; Muldoon, Bridgeford, & Bassett, 2016). Therefore, small-world networks are more likely to represent many psychological phenomena (e.g., psychopathology; Borsboom, Cramer, Schmittmann, Epskamp, & Waldorp, 2011). Moreover, the structure of small-world networks (high clustering and low distances between nodes) is closer to structures produced by factor models than random networks. We manipulated number of variables (10, 20, 30, and 40), density (.20, .40, .60, and

.80), and rewiring probability (.01, .05, .10, .25, and .50). In total, there were 240 conditions (sample size × number of variables × neighborhood × rewiring probability). For each condition, we generated 1,000 samples.

A.2.2 Building Neural Networks Formal articles on steps for how to train neural networks appropriately are sparse; however, there are several resources available. Our approach followed Andrej Karpathy's "recipe" for training neural networks (Karpathy, 2019). This recipe starts by thoroughly inspecting the data distributions and looking for patterns, developing a neural network skeleton by making a simplified model, overfitting a small portion of samples (e.g., 100) from the data, regularizing the model to prevent overfitting (e.g., early stopping), optimizing hyperparameters (e.g., number of nodes and hidden layers, learning rate, batch size), and using neural network ensembles (which we describe in our Introduction section). To prevent overfitting of the training data, we added an *early stopping* criterion: when the validation loss plateaued (i.e., decreases in the loss function less than .001) for ten epochs (or ten runs through the entire training dataset; Prechelt, 2012), then the best weights (highest training accuracy) were kept and used as our model.

A.2.3 Input Nodes Neural networks require a specific structure for input. We used the proportion of loading effect sizes to summarize the covariance matrix into a specific set of variables for input. This approach makes it so that no matter how many variables are in a dataset they can always be summarized into the same variables that are fed into the neural network. Using proportions that are equal to or larger than a certain effect size allows for more continuous cut-offs that reduce some of arbitrariness that is inherent in rule-of-thumb effect sizes.

Following Christensen and Golino's (2021) LCT algorithm, we submitted each dataset to EGA and EFA (using the same number of dimensions estimated by EGA). For both the network and factor loadings, we computed the proportion of loadings that were greater than small (.15 and .40, respectively), moderate (.25 and .55, respectively), and large (.35 and .70, respectively) effect sizes as well as the proportion of loadings that were greater than small effect sizes for dominant and cross-loadings (Christensen & Golino, 2021; Comrey & Lee, 2013). For each dataset, this created 10 proportions in total (five proportions for each loading type) that were used as the base input nodes for all neural networks.

Additional input nodes were created by computing the ratio between the exponent of a base network loading (i.e., small, moderate, large, dominant, and cross) and the exponent of the corresponding base factor loading. To normalize these ratios to be between zero and one (the same range as the proportions), we used min-max normalization using the minimum and maximum possible ratio:

$$\frac{x - \frac{\exp(0)}{\exp(1)}}{\frac{\exp(1)}{\exp(0)} - \frac{\exp(0)}{\exp(1)}},$$

where x is the ratio between the exponent of a network loading proportion (e.g., small) and the exponent of the corresponding factor loading proportion. Additional inputs were not included if they did not increase prediction beyond the base model when training the logistic regression. For all models, only the dominant ratio improved the accuracy of the logistic regression predictions. When additional inputs were used, we mention them in their corresponding neural network descriptions. Below is a figure representing the data processing pipeline to be fed data into the neural network (Figure 4).

Figure 4 displays simulated data from a factor model with two factors, six variables per factor, small correlations between factors (0.30), and sample size of 1000. The pipeline from data to neural network starts by computing a correlation matrix. Next, EGA is used to estimate the number of dimensions. These dimensions are then used as the number of factors to estimate in an EFA model with oblimin rotation. Factor and network loadings are then computed. The proportion of loadings that are greater than or equal to small, moderate, and large effect sizes are computed. Similarly, the proportion of dominant and cross-loadings that are greater than or equal to a small effect size are computed. This is done for both factor and network loadings. Finally, these loadings are fed as input into the neural network. The neural network then predicts the model that the data were generated from. The neural network was visualized using the *nnvizRt* Shiny application (Jessen, 2021) in R.

A.2.4 Activation Function Activation functions determine the output from a node given the input to the node. All hidden layers for all neural networks used the *Parametric Rectified Linear Unit* (PReLu; He, Zhang, Ren, & Sun, 2015) activation function. The *Rectified Linear Unit* (ReLu; Nair & Hinton, 2010) is the contemporary choice for most applications of deep learning (as opposed to the historically and often still used sigmoid function; Nwankpa, Ijomah, Gachagan, & Marshall, 2018). The ReLu activation function is a non-linear function that returns the input of the function as the output unless the input is negative, which is instead set to zero (inspired by the action potential of biological neurons). One limitation of the ReLu function is that it can cause some neurons to never activate (no matter the input), always outputting zero (known as the "dying neuron problem"; He, Zhang, Ren, & Sun, 2015). PReLu overcomes this issue by allowing a trainable parameter α to be adjusted so that some small non-zero negative weights still activate neurons in the network. When α is zero for a node, then PReLu is equivalent to ReLu. This flexibility of PReLu allows it to perform at least as well as ReLu. For all output layers, we used the sigmoid function ($\frac{e^x}{e^x+1}$) to estimate the probability of a given sample belonging to the outcome model (i.e., the model designated as 1 in the output). A cut-off probability of .50 was used to determine what model the sample belonged to (e.g., factor vs. network model).

A.2.5 Gradient Descent Optimizer For all models, we used the *Nestorov Adaptive Moment Estimation* optimizer (NADAM; Dozat, 2016), which tends

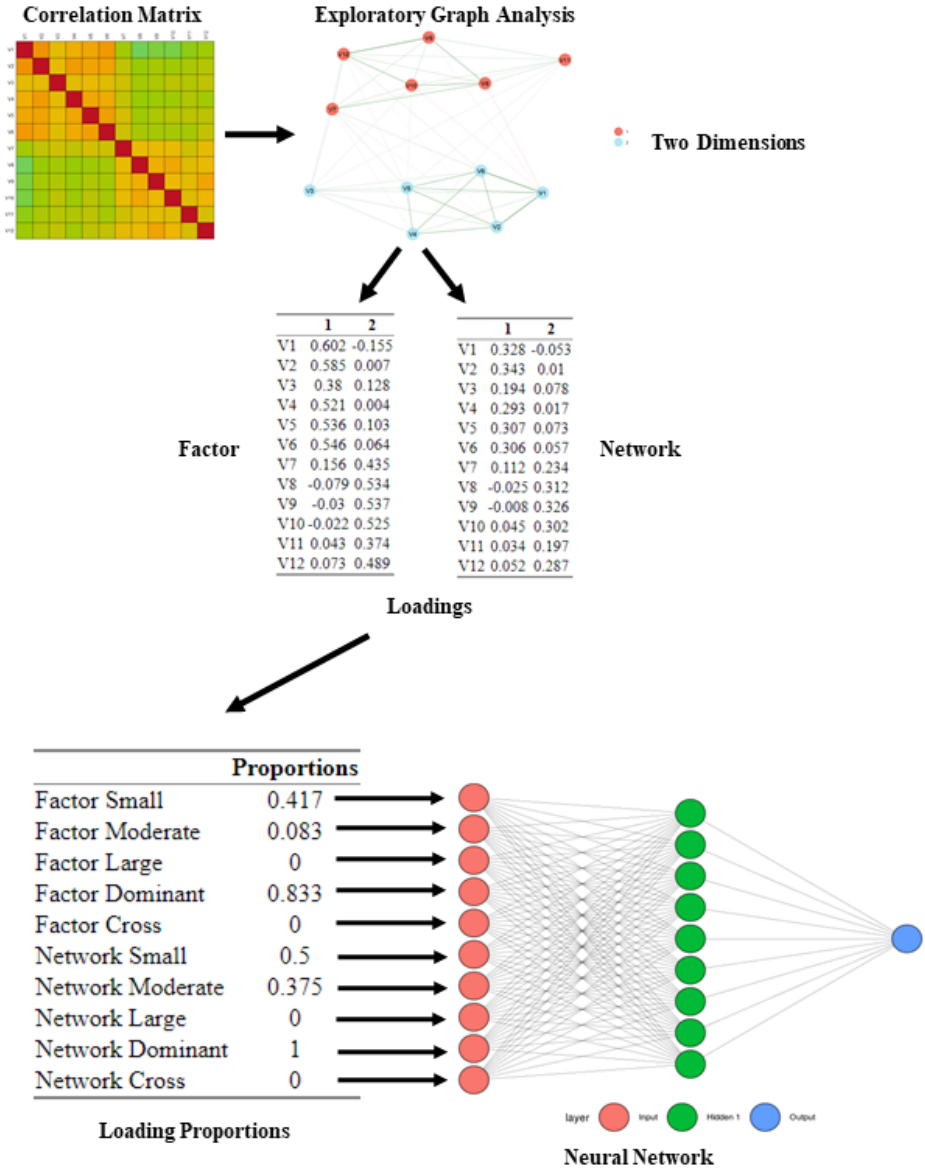


Figure 4. From Data to Neural Network Pipeline

to be the contemporary choice of neural networks (Ruder, 2016). The details of gradient descent optimizers are beyond the scope of this paper; however, their purpose was to minimize their functions by iteratively moving towards the steepest part of the *gradient* or slope of the loss function (Watt, Borhani, & Katsaggelos, 2016). At each iteration, the algorithm takes certain sized steps on the gradient, which are called the *learning rate*. Higher learning rates lead to larger steps toward a loss minimum but can potentially over-step a more optimal minimum; lower learning rates are more likely to reach an optimal minimum but take more time to get there. NADAM is an adaptive algorithm that changes the learning rate over time in order to achieve appropriate descent. The foundation of this algorithm is based on the Adaptive Moment Estimation (ADAM) optimizer (Kingma & Ba, 2014), but uses an alternative momentum parameter called Nesterov's accelerated gradient momentum (NAG; Sutskever, Martens, Dahl, & Hinton, 2013). In NADAM, NAG moves toward an intermediate direction and then corrects toward the gradient, which allows the momentum to be shifted toward the minimum (even after moving past the minimum; for more details, see Dozat, 2016).

A.2.6 Loss and Accuracy Gradient descent optimizers aim to minimize a loss function or the error between the actual and predicted outcomes. In our neural networks, this was *binary cross entropy* or logarithmic loss. Binary cross entropy is defined as the distance between two probability distributions (e.g., actual and predicted outcomes) and mathematically represented as:

$$CE = -\big(y \log(p) + (1 - y) \log(1 - p)\big),$$

where y is the actual model and p is the predicted probability of the dataset belonging to the model. If $y = 1$, then CE; otherwise, if $y = 0$, then $1 - CE$.

Binary accuracy was our accuracy measure, which is the mean of correct identifications in the total sample. The accuracy typically corresponds to loss but not necessarily. This is because correct model identifications are part of the binary cross entropy equation. Their difference arises in the fact that binary cross entropy considers the probability in which a dataset belongs to the correct model. In the random vs. non-random model, for example, a probability $\geq .50$ would be considered a random model (1); otherwise, it is considered a non-random model (0). A correct identification would be a 1 but its probability could be as low as .50. In terms of binary cross entropy, the loss for a correct identification could range from 0 ($p = 1$) to 0.693 ($p = .50$). Therefore, loss is informative about the decisiveness of the predictions and accuracy is informative about the correctness of the predictions.

A.2.7 Training Neural Networks Models were set up with a certain number of samples, which were then split into the original training (80% of the overall sample) and validation (20% of the overall sample) samples. The original validation samples are then completely held out of the model training phase and

were only seen after the model had been trained. The original training samples were used to train the model. During training, the original training samples are further split into a new training dataset (80% of the training samples) and validation dataset (20% of the training samples). This new training dataset is then randomly sampled without replacement with a specific number of *batch sizes* (number of training samples used in each update of the gradient and weights). After all of the new training dataset samples have been used once, the model is tested using the new validation dataset.

Loss and accuracy metrics are then provided with the training loss and accuracy representing the last model in the epoch and the validation loss and accuracy representing the performance of this last model on the validation dataset. The conclusion of a single run of this process is called an *epoch*. Each new epoch will randomly draw samples without replacement from the original training samples and form new training and validation datasets (a process known as *shuffling*). For all neural networks, we set the total number of epochs to 100 to allow training to proceed as necessary to settle into a minimum. Training was terminated when either the epochs reached 100 or our early stopping criterion was reached (i.e., decrease in validation loss less than .001 for ten consecutive epochs). After training was terminated, the final model was then tested on the original validation samples, which are considered to be novel because they had not been seen during the training of the model.

As a baseline comparison model, we trained the lasso regularized logistic regression models on the same original training data using the same input variables used in the neural networks. Regularized logistic regression models were chosen as a comparison for two reasons: (1) logistic regression models tend to perform better than other machine learning classification methods, such as support vector machines and decision trees, when there are overlapping classes, and (2) regularization reduces the flexibility of the model, which makes it less likely to overfit the underlying function in the training data and more likely to generalize to other data conditions (James, Witten, Hastie, & Tibshirani, 2013). The use of logistic regression models provides inference into whether more complex neural networks are necessary. The coefficients of each trained logistic model were extracted and then solved for each case of the original validation dataset. Accuracy and loss were then computed for the original validation dataset.

A.2.8 Feature Importance In order to determine the importance of each input into each neural network, we computed a measure of feature importance on the original validation datasets that were held out of the original training datasets following Fisher, Rudin, and Dominici (2019). The approach works by permutating one-by-one each input variable and computing the loss. The loss is then divided by the original loss to obtain the relative decrease in performance for the permutated input. Because of the stochasticity of the permutations, we computed this analysis ten times and computed the mean of the values. Values greater than one suggest the input was important with larger values suggesting greater importance whereas values near one suggest that the input did not

improve the model and less than one suggest that the input made the model worse.

A.2.9 Data Analysis All analyses were performed in R. All neural networks were trained using the *keras* package (Allaire & Chollet, 2020) and all logistic regression models were fit with the *glmnet* package (Friedman, Hastie, & Tibshirani, 2010). All data, R code and scripts are available on the Open Science Framework (OSF). Each neural network is available on the OSF and can be further fine-tuned and improved with new data and examples (i.e., the models can be further trained with new models, data conditions, and methods of data generation).

A.2.10 Results The mean proportions of the base network and factor loadings across each data-generating models are presented in Figure 5.

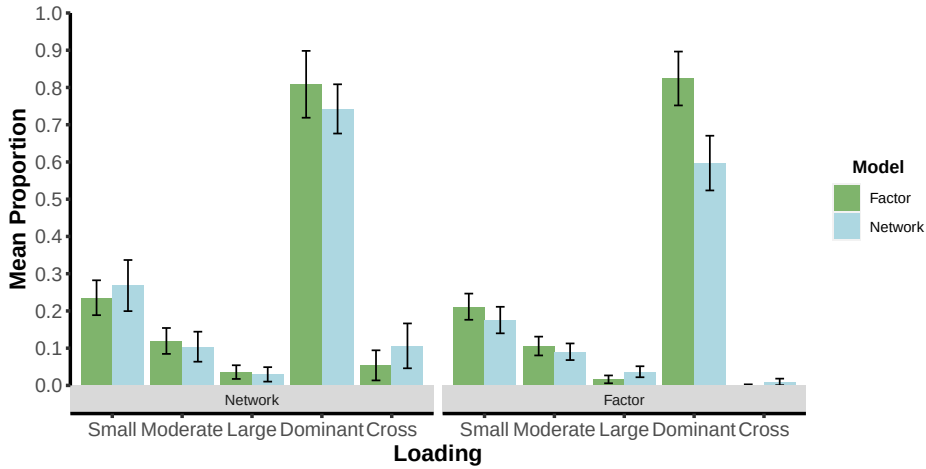


Figure 5. Mean proportions of the base input loadings for the neural network models. Error bars represent 0.5 standard deviations.

The most glaring differences between models are proportion of dominant loadings that achieve a small effect size or larger for both network and factor loadings. This difference is most apparent when the data are generated from a network model. Table 1 presents a summary of the architecture the neural networks including the parameters and validation estimates.

Across all neural networks, we found comparable or better performance than the logistic regression models, suggesting that the neural networks were reasonable and potentially necessary for optimal performance in the LCT algorithm.

Model 1: Low Correlation Factor vs. Network. For the low correlation factor vs. network model, we removed the datasets corresponding to the

Table 1. Neural Network Architectures, Parameters, and Metrics

				Neural Network		Logistic Regression	
Model	Architecture	Batch Size	Learning Rate	Loss	Accuracy	Loss	Accuracy
1	11 9 1	64	.0003	0.180	0.928	0.286	0.890
2	11 9 1	32	.0005	0.159	0.937	0.257	0.908
3	11 9 1	32	.001	0.239	0.902	0.401	0.846

Note. Model: 1 = low correlation vs. network; 2 = high correlation with variables greater than factors vs. network; 3 = high correlation with variables less than or equal to factors vs. network. Grey boxes denote best values of loss and accuracy for each model.

data generated from factor models with correlations between factors of .50 and .70 (120,000 samples), leaving us with 120,000 samples of orthogonal and low correlations between factors (.00 and .30, respectively). To obtain an equivalent number of datasets generated from the network models, we randomly sampled 40,000 network datasets from each level of sample size (i.e., 250, 500, and 1000), resulting in 120,000 total network datasets. In total, we used 240,000 datasets.

We created a single binary output variable with 1 corresponding to a factor model and 0 corresponding to not a low correlation factor model. Importantly, it is possible that the learned weights of the low correlation factor model could still correspond to other factor models even though they weren't observed in the trained model. This potential for overlap was on purpose and allowed multiple checks of factor models to be learned against network models in the LCT algorithm.

The input of this model consisted of our base input nodes along with an additional input: dominant ratio. This made for eleven input nodes in total. There was one hidden layer with nine nodes. Our final model did not reach our early stopping criterion and was terminated after the 100^{th} epoch. We then evaluated the model on the validation dataset, which achieved a loss of 0.180 and accuracy of 92.8%. The neural network model outperformed the regularized logistic regression model by a full tenth in loss and over three percent in accuracy (Table 1). The inputs that had the greatest importance were the cross factor loading (2.57), dominant factor loading (2.55), and large factor loading (2.12).

Model 2: High Correlation with Variables per Factor Greater than Factors vs. Network. The setup of the high correlation with variables greater than factors vs. network model was identical to Model 1 except the samples retained were the high correlation between factors (i.e., .50 and .70; 120,000 samples) rather than the low correlation between factors. From these samples, we extracted samples that were generated with the number of variables per factor that were greater than the number of factors (e.g., 4 variables per factor and 3 factors). This resulted in 60,000 samples used in training. Just as Model 1, we

Table 2. Importance of Input for Each Model

	Network					Factor					Ratio
Model	Small	Moderate	Large	Dominant	Cross	Small	Moderate	Large	Dominant	Cross	Dominant
1	2.09	1.53	1.16	1.59	1.68	1.66	1.21	2.12	2.55	2.57	1.09
2	2.20	2.27	1.23	1.28	1.27	3.07	1.30	2.44	1.54	3.28	1.93
3	6.65	1.81	1.35	2.05	1.88	1.67	1.18	1.84	2.22	2.45	1.37

Note. Model: 1 = low correlation vs. network; 2 = high correlation with variables greater than factors vs. network; 3 = high correlation with variables less than or equal to factors vs. network. Grey boxes denote top three most important features for each model.

randomly sampled an equal number of network samples across the same sample size levels, resulting in a total of 120,000 samples.

The exact same input and hidden layers were used as Model 1. Similarly, our final model did not meet our early stopping criterion and was terminated after the 100$^{\text{th}}$ epoch. We then evaluated the model on the validation dataset, which achieved a loss of 0.159 and accuracy of 93.7%. This neural network outperformed the logistic regression model in loss and accuracy (Table 1). The inputs that had the greatest importance were the cross factor loading (3.28), small factor loading (3.07), and large factor loading (2.44).

Model 3: High Correlation with Variables per Factor Less than or Equal to Factors vs. Network. The setup of the high correlation with variables less than or equal to factors vs. network model was the same as Model 2 except models with the samples retained were the moderate and high correlation between factors (i.e., .50 and .70) with variables per factor than were equal to or less than the number of factors (60,000 samples; e.g., 3 variables per factor and 5 factors). Similarly, an equivalent number of network models were randomly drawn from the 240,000 network models (across the same sample size levels), resulting in a total of 120,000 samples. The inputs and hidden layers were the same as Model 1 and 2. Our final model reached our threshold of early stopping on epoch 88. We then evaluated the model on the validation dataset, which achieved a loss of 0.239 and accuracy of 90.2%. Relative to the other models, the neural network substantially outperformed the logistic regression model on both loss and accuracy (differences of .162 and 5.6%, respectively). The inputs that had the greatest importance were the small network loading (6.65), cross factor loading (2.45), and dominant factor loading (2.22).

A.3 Reproducible Code for the Loadings Comparison Test with Big Five Inventory

```
# Set seed
set.seed(3532)

# Install latest EGAnet package
devtools::install_github("hfgolino/EGAnet")
```

```
# Load packages
library(psych)
library(EGAnet)
library(psychTools)

# Get BFI data
bfi.data  <-  bfi[,1:25]

# LCT of the full dataset
LCT(bfi.data)

# Randomly sample from BFI data
samps  <-  sample(1:nrow(bfi),  nrow(bfi))

# Split samples into sizes of 400
start  <-  seq(1,  nrow(bfi),  400)
end  <-  seq(400,  nrow(bfi),  400)

# New samples
new.samps <- list()

for(i  in  1:length(start)){
        new.samps[[i]] <-
            bfi.data[samps[start[i]:end[i]],]
}


# Apply LCT to new BFI samples
res.bfi <- lapply(new.samps, LCT)

## Empirical
mean(lapply(res.bfi,
  function(x){x$empirical}) == "Factor")

## Bootstrap
mean(lapply(res.bfi,
  function(x){x$bootstrap}) == "Factor")

## Proportion
mean(lapply(res.bfi, function(x){
        names(x$proportion)[which.max(x$proportion)]
}) == "Factor")
```

A.4 Reproducible Code for the Loadings Comparison Test with Default Mode Networks

```
# Install latest EGAnet package
devtools::install_github("hfgolino/EGAnet")

# Load packages
library(googledrive)
library(EGAnet)

# Create path to temporary file
temp <- tempfile()

# Download to temporary file
drive_download( paste("https://drive.google.com/file/d/",
"1T7_mComB6HPxJxZZwwsLLSYHXsOuvOBt", "/view?usp=sharing",
sep = ""), path = temp)

# Load resting state brain data
load(temp)

# Get default mode network from Shen atlas
# (from NetworkToolbox)
atlasNet  <-
c(2,4,3,2,3,3,2,2,2,1,4,1,3,2,4,1,2,4,2,4,2,
2,5,5,5,5,5,4,4,2,2,4,5,5,5,4,5,5,5,5,8,6,
8,4,5,5,2,2,3,3,5,1,1,1,2,1,1,5,8,5,5,5,5,
1,1,8,8,6,8,2,8,6,8,8,6,7,6,7,6,6,7,6,4,5,
3,3,6,4,5,3,4,5,4,4,4,3,5,6,4,7,4,7,4,4,4,
4,4,4,5,4,2,2,4,4,3,2,4,4,4,4,4,4,4,4,4,4,
4,4,4,4,4,4,4,3,4,4,1,3,2,1,3,2,2,4,1,4,2,
1,1,1,1,4,1,2,4,1,2,5,5,5,5,1,5,2,1,5,5,5,
4,5,5,5,5,5,8,6,8,4,5,5,5,2,1,2,1,1,1,5,5,
1,5,1,2,1,5,2,5,6,2,8,8,5,3,8,6,8,6,6,8,8,
6,7,7,7,6,6,4,5,1,4,4,3,3,4,3,4,3,5,4,4,4,
4,4,4,5,4,4,4,3,8,7,2,4,4,4,2,2,4,4,4,4,4,
4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4)

dmn <- which(atlasNet == 3)

# Grab only default mode networks
rest.dmn <- restOpen[dmn,dmn,]

# Convert array to list
dmn.list <- list()

## Make diagonals 1
for (i  in  1:dim(rest.dmn)[3]){
```

```
        net <- rest.dmn[,,i]
        diag(net) <- 1
        dmn.list[[i]] <- net
}

# Apply LCT to DMN list
## 150 = length of original time series
res.dmn <- lapply(dmn.list, LCT,  n  =  150)

## Empirical
mean(lapply(res.dmn, function(x){x$empirical}) == "Network")

## Bootstrap
mean(lapply(res.dmn, function(x){x$bootstrap}) == "Network")

## Proportion
mean(lapply(res.dmn, function(x){
        names(x$proportion)[which.max(x$proportion)]
}) == "Network")
```

A.5 Session Information for Appendix A.3 and A.4

```
R version 4.0.5 (2021-03-31)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19042)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] psych_2.1.3       EGAnet_0.9.9      googledrive_1.0.1
[4] papaja_0.1.0.9997       ggplot2_3.3.3


loaded via a namespace (and not attached):
 [1] pillar_1.6.0      compiler_4.0.5    tools_4.0.5
```

```
 [4] digest_0.6.27     nlme_3.1-152      lattice_0.20-44
 [7] evaluate_0.14     lifecycle_1.0.0   tibble_3.1.1
[10] gtable_0.3.0      pkgconfig_2.0.3   rlang_0.4.11
[13] xfun_0.22         DBI_1.1.1         parallel_4.0.5
[16] yaml_2.2.1        withr_2.4.2       stringr_1.4.0
[19] dplyr_1.0.6       knitr_1.33        generics_0.1.0
[22] vctrs_0.3.8       grid_4.0.5        tidyselect_1.1.1
[25] glue_1.4.2        R6_2.5.0          fansi_0.4.2
[28] rmarkdown_2.8     bookdown_0.22     purrr_0.3.4
[31] magrittr_2.0.1    scales_1.1.1      ellipsis_0.3.2
[34] htmltools_0.5.1.1 mnormt_2.0.2      assertthat_0.2.1
[37] colorspace_2.0-1  utf8_1.2.1        stringi_1.6.1
[40] munsell_0.5.0     tmvnsim_1.0-2     crayon_1.4.1
```

A.6 Example of data-generating model Manipulation

To demonstrate how the structure of data can be manipulated toward a certain model, we used a dataset that consisted of 2,832 observations on items from the Broad Autism Phenotype Questionnaire (BAPQ; Hurley, Losh, Parlier, Reznick, & Piven, 2007) that was a collected as a part of the Simons Foundation Autism Research Initiative's Simplex Collection (`https://www.sfari.org/`). The BAPQ was completed by the mothers and fathers of children on the Autism spectrum. The BAPQ consists of three sub-scales—aloof personality, pragmatic language problems, and rigid personality—that are based on direct assessment interviews with parents of autistic people that correspond to defining behavioral domains of autism: social, communication deficits, and stereotyped-repetitive behaviors (Hurley, Losh, Parlier, Reznick, & Piven, 2007). The BAPQ has demonstrated a robust three-factor structure (Ingersoll, Hopwood, Wainer, & Donnellan, 2011) with each sub-scale containing twelve items that are rated on a 6-point Likert scale. Correlations between the means of the sub-scales tend to be highly correlated in clinical samples (r's from .50 to .70; Hurley, Losh, Parlier, Reznick, & Piven, 2007) but smaller when using factor analysis in non-clinical samples (r's from .10 to .30; Ingersoll, Hopwood, Wainer, & Donnellan, 2011).

Because we have data for both mothers and fathers, we applied the LCT to each parent's datasets. We split the datasets into training (n = 1,699) and testing (n = 1,133) sets to validate the LCT's results. Below we present a table (Table 3) for the predictions of the LCT.

The results demonstrate that the BAPQ in mothers is a factor model based on the empirical prediction and network model based on the bootstrap and proportion prediction. For the fathers, the training data were predicted to be a factor model across all predictions while the testing data were predicted to be a network model across all predictions. In short, the results are mixed but lean towards a network model with three out of for datasets having network predictions for the proportion prediction. Based on this result, we would conclude that the data for mothers and fathers are most likely generated from a network

Table 3.

		Predictions		
Parent	**Dataset**	Empirical	Bootstrap	Proportion
Mother	Training	Factor	Network	Network (0.59) Factor (0.41)
	Testing	Factor	Network	Network (0.71) Factor (0.29)
Father	Training	Factor	Factor	Factor (0.72) Network (0.28)
	Testing	Network	Network	Network (0.55) Factor (0.45)

model. Notably, the fathers' datasets were leaning towards a factor model relative to the mothers datasets (including the training data being a factor model across predictions).

If, for example, we thought that the data generating mechanism was a factor model, then we should try to adjust the data's structure toward a factor model. To do so, we could analyze the structure of the data to see which items are multidimensional or leading to larger cross-loadings between dimensions. One approach for achieving these results is called *bootstrap exploratory graph analysis* (bootEGA; Christensen & Golino, 2019).

bootEGA applies a parametric bootstrap approach where N number of replicate samples are generated from a multivariate normal distribution based on the empirical correlation matrix. Each replicate sample is then analyzed using EGA (see Appendix A.1 for a description), forming a distribution of factors and item placement within those factors. Taking advantage of the deterministic allocation of items into factors, we can determine how often items are replicating in their empirical dimension as well as other dimensions. That is, we can determine how stable the factors are with respect to how items are placed into them (Christensen, Golino, & Silvia, 2020). Items that are not replicating well in their empirically derived factor (e.g., EGA identified factor) indicate that these items are likely to be multidimensional, have larger cross-loadings, and are likely leading the data structure to be more like a network model.

When performing such an analysis, we found that there were four factors with identical item placement for the mothers and fathers datasets' empirically derived structure (using EGA). Using this factor structure and item placement, we applied bootEGA ($n = 500$) to the training and testing datasets for both mothers and fathers. The item stability analysis found one factor containing items that were relative unstable. These items and their stability (number of times replicating in their empirically derived structure) are presented in Table 4. When removing these items, the data structure for all datasets moved closer to a factor model structure as shown in Table 5.

Indeed, three out of four datasets now suggest a factor model relative to a network model. For those three models suggesting a factor model (mothers

Table 4.

Item Description	Replication Proportion			
	Mother		Father	
	Training	Testing	Training	Testing
7. I am "in-tune" with the other person during conversation	0.41	0.59	0.18	0.11
12. People find it easy to approach me	0.33	0.10	0.03	0.02
21. I can tell when someone is not interested in what I am saying	0.42	0.62	0.18	0.11
23. I am good at making small talk	0.33	0.10	0.03	0.02
25. I feel like I am really connecting with other people	0.33	0.10	0.03	0.02
28. I am warm and friendly in my interactions with others	0.34	0.10	0.03	0.02
34. I can tell when it is time to change topics in conversation	0.42	0.62	0.18	0.11

Table 5.

Parent	Dataset	Predictions		
		Empirical	Bootstrap	Proportion
Mother	Training	Factor	Factor	Factor (0.73) Network (0.27)
	Testing	Network	Network	Network (0.58) Factor (0.42)
Father	Training	Factor	Factor	Factor (1.00) Network (0.00)
	Testing	Factor	Factor	Factor (0.72) Network (0.28)

training and both fathers), all predictions were for a factor model. The testing mothers dataset was a network across all predictions but notably the proportions prediction suggested that the model moved away from a network model and closer to a factor model (from 0.71 to 0.58 for a network model and 0.29 to 0.42 for a factor model).

Journal of Behavioral Data Science, 2021, *1* (1), 127–153.
DOI: https://doi.org/10.35566/jbds/v1n1/p6

A Multiple Imputation Approach for Handling Missing Data in Classification and Regression Trees

Danielle M. Rodgers[1], Ross Jacobucci[2], and Kevin J. Grimm[1]

[1] Arizona State University, Tempe, AZ 85281, USA
dmrodge3@asu.edu, kjgrimm@asu.edu
[2] University of Notre Dame, Notre Dame, IN 46556, USA
rjacobuc@nd.edu

Abstract. Decision trees (DTs) is a machine learning technique that searches the predictor space for the variable and observed value that leads to the best prediction when the data are split into two nodes based on the variable and splitting value. The algorithm repeats its search within each partition of the data until a stopping rule ends the search. Missing data can be problematic in DTs because of an inability to place an observation with a missing value into a node based on the chosen splitting variable. Moreover, missing data can alter the selection process because of its inability to place observations with missing values. Simple missing data approaches (e.g., listwise deletion, majority rule, and surrogate split) have been implemented in DT algorithms; however, more sophisticated missing data techniques have not been thoroughly examined. We propose a *modified multiple imputation approach* to handle missing data in DTs, and compare this approach with simple missing data approaches as well as single imputation and a multiple imputation with prediction averaging via Monte Carlo Simulation. This study evaluated the performance of the missing data approaches when data were missing at random or missing completely at random. The proposed multiple imputation approach and the surrogate split approach had superior performance with the proposed multiple imputation approach performing best in the more severe missing data conditions. We conclude with recommendations for handling missing data in DTs.

Keywords: Multiple Imputation · Classification and Regression Tree (CART) · Missing Data

1 Introduction

Missing data are endemic in research and appropriate handling of missing data is required to ensure unbiased parameter estimates. Missing data are often caused

by participant nonresponse due to an unwillingness to divulge information, inadvertent skipping, fatigue, or time considerations (Hattie, 1983; Holmanx& Glas, 2005; Huggins-Manley, Algina,x& Zhou, 2018; Moustakix& Knott, 2000). Missing data are particularly problematic when nonresponding participants systematically differ from participants who completed the study. Known as nonresponse bias (Lavrakas, 2008), systematic differences in responding may affect estimated model parameters and threaten the validity of conclusions drawn from the statistical model (Enders, 2010; Grovesx& Peytcheva, 2008; Lavrakas, 2008).

Several methods have been developed for handling missing data due to nonresponse (Baraldix& Enders, 2010). One widely recommended approach for handling missing data is multiple imputation (Allison, 2002; Baraldix& Enders, 2010; Enders, Dietz, Montague,x& Dixon, 2006; Schaferx& Olsen, 1998). Multiple imputation is a four-step procedure. First, plausible values from a distribution specifically modeled for the missing data are drawn. Second, the statistical model is fit to the imputed dataset and parameter estimates and standard errors are retained. Third, the first two steps are repeated a specified (e.g., 20) number of times. Fourth, the parameter estimates and standard errors are pooled to determine the point estimate for each parameter along with an appropriate standard error (Enders, 2010; Rubin, 1987; van Buurenx& Groothuis-Oudshoorn, 2011). Proper standard errors are calculated to account for the within (square of the average standard error) and between (variance of the parameter estimates across imputations) imputation variation in the parameter estimates. This final step is referred to as the *pooling step*.

Multiple imputation is an effective missing data strategy for theoretically-driven statistical models (e.g., regression, ANOVA, etc.; Baraldix& Enders, 2010); however, the *pooling step* can be challenging when fitting exploratory/data driven models because the statistical models for each imputed dataset may include different model parameters (i.e., due to variable selection). Decision trees (DTs) are an exploratory model where the standard multiple imputation approach is not viable. In DTs, the data are recursively split into two nodes based on the variable and value that lead to an optimal prediction. Implementing the standard multiple imputation approach with DTs will likely lead to different variables being selected to partition the data in each imputed dataset, which makes the *pooling stage* challenging, if not impossible. In this paper, we propose and examine the performance of a *modified multiple imputation approach* for handling missing data with DTs. We compare the performance of the proposed approach against the standard approach for handling missing data in DTs (*surrogate splits*), simple missing data approaches (*listwise deletion*, *delete if selected*, and *majority rule*), single imputation, and a multiple imputation approach that ignores variation DT structures and pools the predicted values from the DTs (*multiple imputation with prediction averaging*). We continue with an overview of the classification and regression tree (CART) algorithm for DTs, review currently implemented missing data approaches in CART, and describe our proposed multiple imputation approach. We then outline and review our

simulation study to evaluate the performance of each missing data approach, and conclude with recommendations.

1.1 Classification and Regression Tree (CART)

CART is an algorithm for DTs that has become a very popular machine learning technique because of its ability to create powerful prediction models with non-linear and interactive effects. Moreover, the resulting DT is easy to interpret. CART is a greedy DT algorithm that recursively partitions the data and considers the mean (quantitative outcome) or the mode (categorical outcome) as the predicted value within each partition (James, Witten, Hastie,x& Tibshirani, 2013; Loh, 2011). Three critical aspects of the CART algorithm are *variable splitting* (*fit criteria*), *stopping criteria*, and *model selection*. For variable splitting, the CART algorithm selects the variable and partitioning value that splits the data into two groups where the outcome is maximally homogenous within each group (Breiman, Friedman, Stone,x& Olshen, 1984). The two resulting groups are often referred to as *child nodes* (with the node that was split referred to as the *parent node*). All values of the predictors are considered potential splitting values to partition the data into two child nodes. For a regression tree (numeric outcome), the predictor variable and splitting value that minimizes the residual sum of squares is selected to split the node (Gonzalez, O'Rourke, Wurpts,x& Grimm, 2018; Loh, 2011). For a classification tree (categorical outcome), the predictor variable and splitting value that minimizes the Gini Index (entropy-/information can be used instead of the Gini Index) is selected to partition the node. This process in repeated on each child node until a stopping criterion is reached. Stopping criteria include a minimum improvement in prediction accuracy, tree depth, and sample size required to partition a node. These stopping criteria prevent further node splits, but are not often used for model selection. Once a stopping rule is reached for each node and tree growth has stopped, the DT is then pruned (reduced in size) with the final model selected based on k-fold cross-validation. A large DT is often grown in order to ensure that a useful split is not inadvertently missed because of an arbitrary stopping rule (Breimanxet al., 1984).

1.2 Missing Data Mechanisms

Missing data occur when an observation contains no value for a given variable. There are numerous situations that lead to missing data, which makes it difficult to know exactly how and why each missing value appears in a dataset. Rubin (1976) proposed using observed variables to predict the occurrence of missing values and coined the term *missing data mechanisms* to classify relationships between missing values and the observed variables in a dataset. Specifically, missing data mechanisms describe how the propensity for a missing value relates to other measured variables and itself. Rubin (1976) presented three types of missing data mechanisms: *missing completely at random* (MCAR), *missing at random* (MAR), and *missing not at random* (MNAR). Data are MCAR when

missing values on variable x are unrelated to both the observed variables and the underlying values of x itself (Enders, 2003; Rubin, 1976). Thus, MCAR indicates the occurrence of missing data is purely random making MCAR desirable; however, MCAR assumptions are rarely met in practice (Enders, 2010; Muthén, Kaplan,x& Hollis, 1987; Raghunathan, 2004). Data are MAR when missingness is systematic and correlated with other variables in the dataset. Specifically, data are considered MAR when the missing values on the variable x are related to other variables in a dataset but not related to x itself (Enders, 2003; Rubin, 1976). Most advanced missing data handling procedures (e.g., multiple imputation, full information maximum likelihood) rely on MAR assumptions. Data are MNAR when missing values on x are dependent on the underlying values of x itself (Enders, 2003; Rubin, 1976). Missingness does not depend only on observed data when data are MNAR making it the most challenging missing data mechanism to handle in practice.

The missing data mechanisms determine how well a given missing data approach will perform. According to Baraldixand Enders (2010), deletion approaches (i.e., listwise, pairwise, etc.) perform well in situations when data are MCAR, whereas more advanced approaches, such as multiple imputation or full information maximum likelihood (FIML), outperform deletion and produce unbiased parameter estimates when data are MCAR or MAR. It is important to note that many approaches commonly used to treat missing data (e.g., deletion, imputation, FIML etc.) do not perform well when data are MNAR.

1.3 Missing Data in CART

Missing data are problematic in CART because an observation with a missing value on the predictor variable provides no information about the child node to which the observation belongs. The advanced missing data techniques for handling MAR data, such as multiple imputation and full information maximum likelihood, cannot be applied in a straightforward manner in CART, and DTs more generally. Given the challenges for advanced missing data approaches, simpler strategies have been utilized in CART. We review these approaches next.

1.3.1 Listwise Deletion A simple missing data strategy for CART is to remove observations where a missing value is present. This approach is taken when preparing the data for analysis.

1.3.2 Delete if Selected The second missing data strategy for CART is to retain participants with missing values until a variable with missing values is selected. For example, a participant has a missing value on x_1. This participant would be retained in the DT until x_1 is selected to partition the data. Thus, if x_1 is not selected, then the participant is retained in the model. Importantly, the participant contributes to the formation of the DT until s/he cannot be placed into a child node because of the missing value.

1.3.3 Majority Rule In majority rule, if a variable is selected for partitioning and a participant has a missing value, then the participant is placed in the child node that contains the most observations. Thus, the participant contributes to the formation of the DT even after the participant has a missing value for a selected splitting variable.

1.3.4 Surrogate Splits When an observation has a missing value on a selected splitting variable, surrogate splits uses another variable in the dataset to place the observation into a child node. That is, a *surrogate* variable is used to determine the child node for the observation with a missing value. To do this, the partitioning algorithm is applied with the two child nodes as a classification outcome and the other variables in the dataset as splitting variables (Therneaux& Atkinson, 2019). The usefulness of each surrogate variable is determined by examining the misclassification error for each variable (misclassification error for predicting child node using participants with available data). Additionally, the misclassification rate is computed for *majority rule*, where observations with missing values on the splitting variable is placed in the child node with the most observations. Each variable that performs better than *majority rule* is considered a surrogate and is ranked based on its performance. The first-ranked surrogate variable is then used to place observations with missing values. If an observation is missing the first-ranked surrogate, then the second-ranked surrogate variable is used to place the observation, and so forth. In the rare situations where no surrogate variables are present, the observation is placed in whichever child node contains the most observations (Therneaux& Atkinson, 2019).

1.3.5 Single Imputation Imputation strategies use information from the complete data to estimate what a missing value *would be* if it was observed. Single imputation draws a plausible value from a predictive distribution based on available data (Littlex& Rubin, 2002) to fill in a given missing value. The imputation model is typically built on a linear or logistic regression model depending on the nature of the variable with the missing values; however, imputation models have been built upon more complex algorithms, such as DTs and random forests (Tangx& Ishwaran, 2017). Once data are imputed, the CART algorithm can be implemented using the imputed dataset, which does not have any missing values.

1.3.6 Multiple Imputation with Prediction Averaging Multiple imputation with prediction averaging (Feelders, 1999; Twala, 2009) follows a fairly straightforward multiple imputation approach involving the four steps described above. First, missing values are imputed from a distribution specifically modeled for the missing data. Second, a DT is fit to the imputed data. Third, the first and second steps are repeated multiple (e.g., 20) times. Fourth, the predicted values from the DTs for an individual are averaged and the average serves as the predicted value for the individual. This approach does not try to summarize

the decision rules of the DTs – just their predicted values. Thus, there is not a single DT with a single set of decision rules that can be interpreted. Averaging predicted values from the DTs fit to multiple imputed datasets is a viable approach when researchers are primarily interested in prediction because of the lack of interpretability. This approach will likely lead to better prediction accuracy because it is similar to *bagging* (Breiman, 1996).

1.3.7 Comparative Studies Several studies have been conducted to compare DT missing data approaches (Batistax& Monard, 2003; Beaulacx& Rosenthal, 2020; Feelders, 1999; Twala, 2009). Across four studies, the following missing data approaches have been evaluated: listwise deletion, surrogate splits, single imputation (i.e., k-nearest neighbor imputation, mean/mode imputation, EM/logistic imputation, decision tree imputation, and distribution based imputation), multiple imputation with prediction averaging, separate class, Branch-Exclusive Splits Tree (BEST) algorithm, and several methods that were developed and implemented in other DT algorithms (e.g., `C4.5` and `C5.0`). Nearly all studies used complete data sets (from the UCI machine learning repository) and artificially imposed missing values.

The studies that evaluated multiple imputation with prediction averaging found this approach outperformed all approaches it was compared against (e.g., single imputation, surrogate splits, listwise deletion) when data were MCAR and MAR (Feelders, 1999; Twala, 2009). The same studies found single imputation to be the second-best performing approach (Feelders, 1999; Twala, 2009). However, it is important to consider the different single imputation techniques. For example, EM single imputation performed well for numeric variables (Twala, 2009), whereas decision tree single imputation and k-nearest neighbor imputation performed best with categorical variables (Batistax& Monard, 2003; Twala, 2009). Surrogate splits performed well when there are high correlations among variables (Twala, 2009) and listwise deletion generally performed poorly (Twala, 2009). Separate class and the BEST algorithm approaches have been found to perform well when data were MNAR (Beaulacx& Rosenthal, 2020).

Previous research supports the current method of employing multiple imputation in DTs (i.e., averaging predicted values over different imputed tree structures) when data are MAR or MCAR, but only when a researcher is interested in prediction accuracy and not interested in interpretability. The purpose of this study is to modify the current multiple imputation approach in such a way that the proposed approach produces interpretable tree structures and reduces prediction accuracy inflation.

1.4 Proposed Modified Multiple Imputation Approach

The modified multiple imputation approach for CART follows the first three steps of multiple imputation; however, the pooling step is different. First, data are imputed from a distribution specifically modeled for the missing data. Second, a CART is fit to the imputed data with the *complexity parameter* (*cp*)

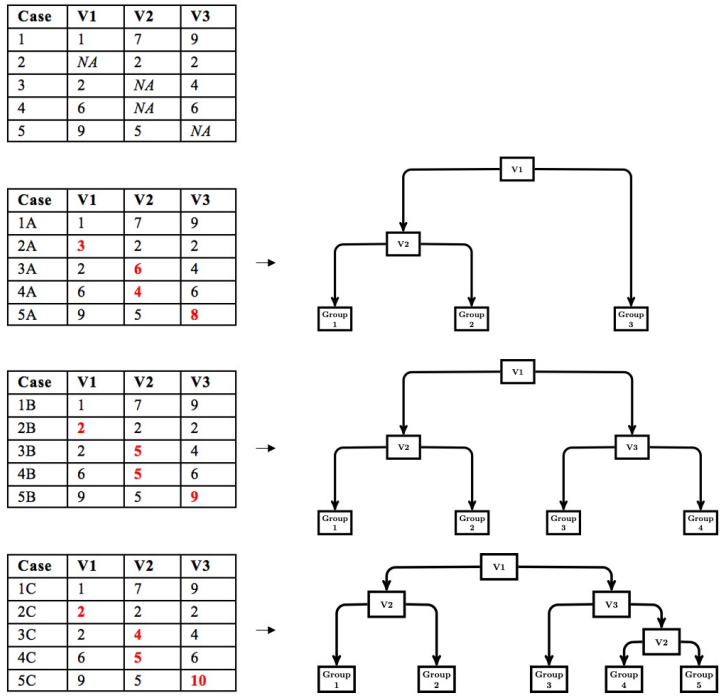


Figure 1. Imputation and analysis phase of the modified multiple imputation approach for DTs.

optimized using cost-complexity pruning through k-fold cross-validation. The *cp* controls the tree size for each imputed dataset. We use the *cp* to control tree size because the *cp* is used in `rpart` (Therneaux& Atkinson, 2019), a common CART package available in R and the package we use in our simulation work. Other measures of tree size (e.g., depth) could be implemented based on availability. Third, the first two steps are repeated multiple times (e.g., 20). Figure figure1 depicts a simple example of the first three steps. Fourth, the imputed datasets are stacked to create a single, large data set consisting of $m \cdot N$ rows, where m is the number of imputed datasets and N is the sample size for each imputed dataset. A CART is then fit to the stacked dataset with the *cp* set to the average of the optimized *cp* obtained when a CART was fit to each imputed dataset. Thus, in this pooling step, we pool the *cp* that controls tree growth and then use this value to fit a CART to the stacked data. This leads to a single DT that is indirectly optimized to the stacked multiply imputed dataset with a single set of decision rules that are easily interpreted (shown in Figure figure2).

Fitting the final CART to the stacked multiply imputed dataset provides an optimal set of decision rules, but ignores the variability across imputed datasets. While imputation variability is an important component of the calculation of standard errors in the application of multiple imputation with a theoretically

Case	V1	V2	V3
1	1	7	9
2	*NA*	2	2
3	2	*NA*	4
4	6	*NA*	6
5	9	5	*NA*

Case	V1	V2	V3
1A	1	7	9
2A	3	2	2
3A	2	6	4
4A	6	4	6
5A	9	5	8
1B	1	7	9
2B	2	2	2
3B	2	5	4
4B	6	5	6
5B	9	5	9
1C	1	7	9
2C	2	2	2
3C	2	4	4
4C	6	5	6
5C	9	5	10

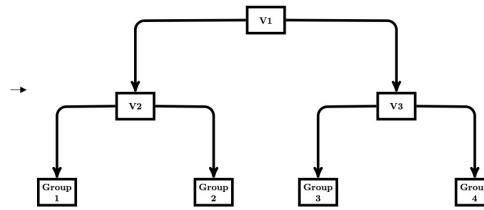


Figure 2. The pooling phase of the modified multiple imputation approach. Multiply imputed datasets are stacked into a single data frame, a DT is fit to the stacked dataset, and the DT is pruned based on the average tree structure from individual DTs.

driven statistical model (e.g., multiple regression model), standard errors are not part of CART (and DTs more generally). The splitting values in CART are considered point estimates and CART does not provide information on the uncertainty of the point estimate.

Pooling the *cp* to control tree size is an important aspect of the modified multiple imputation approach. We note that the optimal *cp* cannot be determined through k-fold cross validation of the stacked multiply imputed data because the different folds of the data are too similar. For example, say we have a dataset with 10% MCAR missingness on ten variables. We conduct $m = 20$ imputations and stack the multiply imputed data. Approximately 35% of the sample will have complete data leading to the same data appearing in the stacked data 20 times. Another 39% of the sample will be missing one value leading to 90% of their data appearing in the stacked data 20 times. The high degree of the same data appearing in the dataset is problematic for k-fold cross-validation because the data from $k-1$ folds that are used to train the algorithm are too similar to the data in the k^{th} fold that is used to test the model. Thus, using k-fold cross validation with the stacked multiply imputed data leads to an overgrown (overfit) CART. Determining tree size based on pooling the *cp* leads to more appropriately sized DTs.

Next, we conduct a Monte Carlo simulation study to examine the performance of the modified multiple imputation approach outlined above and compare its performance to the missing data methods currently implemented with DTs

in terms of its predictive performance, variable selection, variable importance, and tree size.

2 Methods

A Monte Carlo simulation study was conducted to compare how well the different missing data approaches performed with CARTs. Data were generated from a population tree structure, missing values were generated following different missing data protocols, CARTs were fit to these datasets using each missing data handling approach, and we examined various indices of the resulting prediction model. This process was repeated 1,000 times for every condition. Baseline measures were taken from complete datasets (i.e., containing no missing values) and used for comparison. We examined the performance of each missing data approach with respect to prediction accuracy, variable selection, and variable importance. All programming scripts are contained on the third author's website.

2.1 Data Generation

Data were generated using `R` (R Core Team, 2020). All predictor variables were independently drawn from a standard normal distribution (i.e., μ=0, σ=1). Depending on the condition, one (x_1) or four (x_1, x_2, x_3, x_4) variables were created. Three predictor variables, z_1, z_2, and z_3, were then generated to either correlate .4 or .6 with the x variables, and z_1, z_2, and z_3, were subsequently used to generate the outcome using a series of decision rules from a population DT. The population tree structure included six splits and seven terminal nodes. The outcome variable, y, was generated from the population tree shown in Figure figure3 with values generated from a normal distribution with the mean and variance reported in each terminal node. Of note, the first split in the population tree was on z_1. Additionally, six distractor predictor variables, z_4 through z_9 were generated from a standard normal distribution and correlated .15 with z_1, z_2, and z_3. Each simulated dataset included 10 or 13 predictor variables (i.e., three used in the population DT, one or four used for missing data generation, and six distractors), and the outcome variable.

2.2 Manipulated Features

Manipulated features included sample size and characteristics of missing values. The sample sizes considered were $N = 200$, N= 500, or N= 1,000 to cover a range of sample sizes common in the social and behavioral sciences. Missing values were imposed across all predictors, but they were not imposed on the outcome variable. The nature of the missing values only varied for z_1, which was the first splitting variable in the population tree structure. The missing data mechanism was varied, the percentage of missing data, the number of variables that the likelihood of a missing value was dependent on, and the degree of association

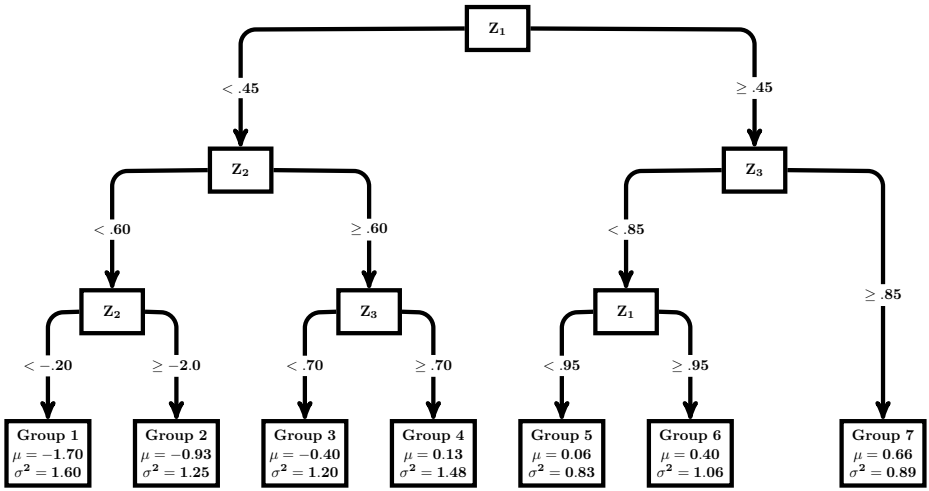


Figure 3. Population Tree Structure

between the likelihood of missingness and the other variable(s) in the dataset. Missing data generation on all other predictors (all predictor variables excluding z_1) were MCAR with a 2.5% probability of being recorded as missing.

2.2.1 Missing Data Generation The method for imposing missing values on z_1 closely followed methods from Mazza, Enders,xand Ruehlman (2015). Missing values were designed to either be missing at random (MAR) or missing completely at random (MCAR). In the MAR condition, missing values on z_1 were generated to relate to one (x_1) or four variables (x_1, x_2, x_3, and x_4). The association between the likelihood of missingness and the other variable(s) in the dataset was specified using a logistic regression model (Agresti, 2012; Johnsonx& Albert, 1999; Mazzaxet al., 2015), with slope and intercept parameters chosen to produce the desired level of association between the underlying missingness probability and the complete variable(s) as well as the overall percentage of missing values. Slopes were selected such that the strength of association between the underlying missingness probability and the complete variable(s) was either $R^2 = .2$ for a moderate association or $R^2 = .4$ for a strong association. Intercepts were selected so that the percentage of missing values on z_1 was either 15% or 30%, which are rates commonly found in psychological and educational research (Enders, 2003). The MCAR condition had fewer manipulated features than the MAR conditions because missingness was unrelated to any other variables in the dataset. Since MCAR occurs when the likelihood of missingness occurs at random, the slope for the logistic regression model was 0 and intercepts were chosen such that the percentage of missing values was either 15% or 30% on z_1.

2.2.2 Approaches for Handling Missing Data Listwise deletion, delete if selected, majority rule, surrogate splits, single imputation, multiple imputation with prediction averaging, and the proposed multiple imputation approach were used to handle the missing data. Listwise deletion was employed by deleting cases with missing values prior to analyses. Delete if selected was applied using the control settings (i.e., `usesurrogate=0`) from the `rpart` package (Therneaux& Atkinson, 2019) in `R` (R Core Team, 2020). Majority rule was also employed using the control function by specifying that no surrogates would be used in the analyses (i.e., `maxsurrogate=0`). By setting the max number of surrogates in the analysis to zero (`maxsurrogate=0`), the algorithm was forced to assign cases with missing values based on majority rule. Delete if selected control setting specifies that the surrogate split method would not be used to treat missing data (`usesurrogate=0`). The surrogate split approach used the default method (i.e., `usesurrogate=2`) to place observations with missing values. If no surrogates were found, then majority rule was enacted.

For single and multiple imputation, data were imputed using the *Multivariate Imputation by Chained Equations* (`mice`) package (van Buurenx& Groothuis-Oudshoorn, 2011) in `R` (R Core Team, 2020). The elementary imputation method was specified using program defaults, which used predictive mean matching. In the single imputation approach, missing values were imputed once to create a single dataset (i.e., $m = 1$), which was then analyzed. In the multiple imputation approaches, missing values were imputed 20 times (i.e., $m = 20$). According to van Buurenxand Groothuis-Oudshoorn (2011), `mice` assumes that the multivariate distribution of an incomplete variable is completely specified by a vector of unknown parameters, θ. Sampling iteratively, the algorithm models the conditional distributions of the incomplete variable given the other variables to obtain a posterior distribution of θ. Using Gibbs sampling, the algorithm selects and fills in plausible values for the missing values on the incomplete variables. Outcome distributions are assumed for each variable instead of the whole dataset. The chained equations within `mice` refer to concatenating univariate procedures to fill in missing data (van Buurenx& Groothuis-Oudshoorn, 2011).

2.2.3 Stopping Criteria DTs recursively partitions data until one of the stopping criteria is reached for each node. Optimal tree sizes were determined using a two-step procedure for listwise deletion, delete if selected, majority rule, surrogate splits, and single imputation. First, all stopping criteria were set to small values to generate an overgrown tree. For all splits in this overgrown tree, 10-fold cross-validation was used to determine the relative cross-validation prediction error associated with the split. The tree was then pruned by specifying the *cp* associated with the smallest estimate of cross-validated prediction error from the 10-fold cross-validation. In multiple imputation, each imputed dataset was analyzed separately and each tree was overgrown. The *cp* associated with the optimal tree size determined through 10-fold cross-validation was retained. In multiple imputation with prediction averaging, the predicted values from each pruned tree were averaged. In the modified multiple imputation approach, the

multiply imputed data were stacked and analyzed with the *cp* set to the average value of the *cp* obtained when the CART was fit to each imputed dataset separately.

There are several viable approaches to choosing tuning parameters in machine learning. This study used the minimum cross-validated prediction error to determine the best model that would optimize prediction accuracy. However, it is important to note that methods like the "one standard error" rule (Breimanxet al., 1984) are often used in practice. The "one standard error rule" uses the most parsimonious model whose error is no more than one standard error above the error of the best model (Hastie, Tibshirani,x& Friedman, 2009).

2.3 Evaluation Metrics

Four evaluation metrics were examined to assess and compare the performance of the missing data approaches. The metrics were the averaged mean square error (MSE) in a test dataset, the proportion of replicates where the first splitting variable was z_1, variable importance metrics, and the median number of splits.

The final DT from each missing data approach was used to generate predicted values in the test dataset with $N = 10{,}000$ drawn from the same population. The test dataset contained no missing values, and was not used to estimate any of the models. The predicted values in the test dataset were calculated and used to determine the MSE - a measure of prediction accuracy. Lower MSE values indicated stronger prediction accuracy, whereas higher MSE values indicated weaker prediction accuracy. The performance of missing data approaches was compared to each other and with the CART estimated using the complete data.

The second evaluation metric was proportion of replicates where z_1 was the first variable selected to split the data. Recall that variable z_1 was the first spitting variable in the population tree. Thus, the proportion of times z_1 (i.e., the target variable) was correctly selected for the first split indicates the CART properly selected the primary splitting variable. The third evaluation metric was variable importance. Variable importance assesses the degree to which each variable contributes to the prediction of the outcome. Variable importance is calculated for every predictor by summing together the decrease in error for every split using the variable as the splitting variable. We assessed and compared variable importance values for z_1, z_2, and z_3 across each missing data approach, and compared variable importance to the values obtained when analyzing the complete data.

The median number of splits was the last evaluation metric. Seven decision trees were fit (i.e., complete data and the six missing data approaches) for each replication within a condition. The median number of splits across all replications within a condition was recorded for each approach. This was compared across missing data approaches and compared to the number of splits in the population DT as an indication of proper tree size.

3 Results

3.1 Summary

Overall, the proposed multiple imputation approach and surrogate splits performed well across all outcome measures. The proposed multiple imputation approach (closely followed by single imputation) performed best when data were MAR with multiple variables strongly predicting missing values and strong associations among predictors. Surrogate splits performed well when data were MCAR or MAR with a single variable predicting missing values and weak associations among predictors. Other approaches stood out on specific outcomes. For example, multiple imputation with prediction averaging had the greatest prediction accuracy. Listwise deletion correctly selected z_1 for the first split more often than all other approaches. However, these methods only performed well on specific outcomes and not across all outcome measures. The following sections summarize and compare the approaches for each outcome.

3.2 Mean Square Error (MSE)

MSE values for each missing data approach are shown in Figure figure4 for four representative conditions. The conditions were selected to represent (1) a mild MCAR condition (i.e., 15% missingness and predictors correlated .16), (2) a mild MAR condition (i.e., 15% missingness, weak association among predictors and missing values ($R^2 = .2$), a single predictor of missingness, predictors correlated .16), (3) a moderate MAR condition (i.e., 30% missingness, greater association among predictors and missing values ($R^2 = .4$), a single predictor of missingness, predictors correlated .36), and (4) a severe MAR condition (i.e., 30% missingness, greater association among predictors and missing values ($R^2 = .4$), multiple predictors of missingness, predictors correlated .36).

Overall, a higher percentage of missing data led to higher MSE across all approaches for handling missing data. This effect was greater in the smaller sample size conditions. Multiple imputation with prediction averaging produced the least amount of bias, which was likely because this approach is an ensemble-type approach like bagging (Breiman, 1996). The average MSE for this approach most closely resembled the results when the CART was fit to the complete data (see Figure figure4). The proposed multiple imputation approach and surrogate splits produced more bias than the multiple imputation approach with prediction averaging. Differences between the proposed approach and surrogate splits were minimal (i.e., average MSE typically only differed by .01) and became less apparent in the larger sample size conditions. The proposed multiple imputation approach produced less bias than surrogate splits when there were multiple predictors of missingness, stronger associations between predictors and missingness, and a higher percentage of missing data (fourth panel in Figure figure4). This approach generally handled small sample sizes ($N = 200$) better than surrogate splits across all MAR conditions.

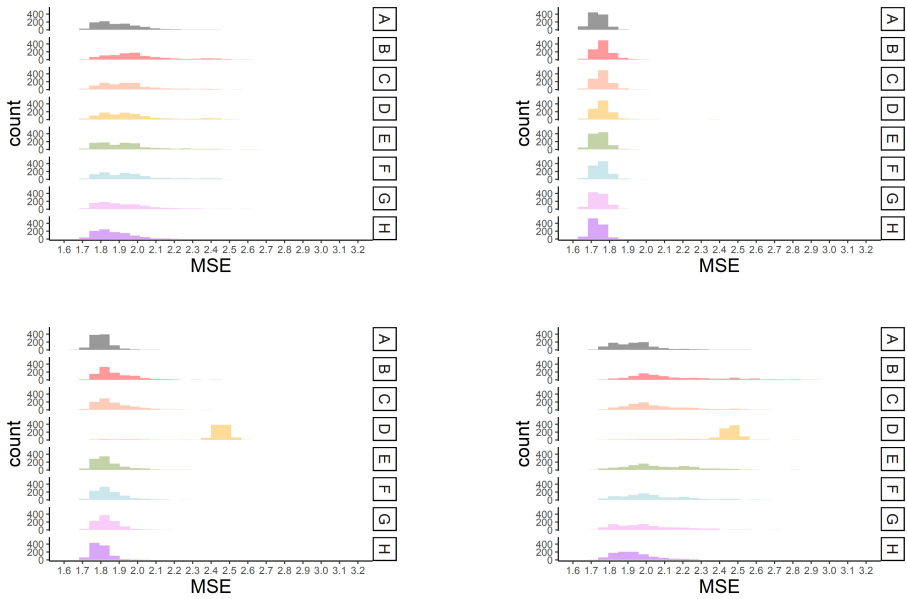


Figure 4. Bias produced in each missing data approach in four representative conditions. Missing data approaches include: (A) Baseline - No Missing Data; (B) Listwise Deletion; (C) Delete if Selected; (D) Majority Rule; (E) Surrogate Splits; (F) Single Imputation; (G) Proposed Multiple Imputation Approach; (H) Multiple Imputation with Prediction Averaging. The first panel represents a condition where 15% of the data on z_1 were MCAR, the predictors were correlated .16, and $N = 200$. In the condition represented in the second panel (top right), 15% of the data on z_1 were MAR with a single variable predicted missing values ($R^2 = .2$), predictors were correlated .16, and $N = 1{,}000$. Third panel (bottom left) represents a condition where 30% of the data on z_1 were MAR where a single variable predicted missing values ($R^2 = .4$), predictors were correlated .36, and $N = 500$. The fourth panel represents a condition where 30% of the data on z_1 were MAR with multiple variables predicted missing values ($R^2 = .4$), predictors were correlated .36, and $N = 200$.

Surrogate splits often produced the same amount of bias as the proposed multiple imputation approach when data were MCAR and in the MAR conditions with a single predictor of missingness and weaker associations between variables and missingness (first and second panel in Figure figure4). Overall surrogate splits produced less bias than the proposed approach across these mild missing data conditions (see Table S1 in supplemental materials). Single imputation closely followed the proposed multiple imputation approach and surrogate splits but had slightly greater average MSE values. Also, single imputation performed fairly well in the conditions where missingness was related to multiple predictors. Delete if selected and listwise deletion produced slightly greater bias across all the conditions and majority rule produced the greatest amount of bias across all conditions.

3.3 Proportion of Correct First Variable Splits

The proportion of times that z_1 was chosen for the first split was recorded. Figure figure5 illustrates the performances of each approach in four example conditions that range from mild to severe missing data conditions in this simulation. Across all approaches, higher rates of missing values led to fewer instances that z_1 was chosen for the first split. Greater effects were found in small sample sizes. Conditions represented in Figure figure5 have a consistent sample size and rate of missing to simplify comparisons across missing data patterns and associations.

Listwise deletion correctly selected first split more frequently than the other approaches and most closely resembled the complete data conditions (see Figure figure5). The performance of the other approaches depended on the missing data pattern, strength of association among predictors and missing values, and the percentage of missing data. When data were MCAR, surrogate splits and delete if selected correctly chose z_1 for the first split more often than the remaining approaches (first panel in Figure figure5).

Performance across the MAR conditions depended on the strength of association among predictors and percent missingness. When there were weak associations between predictors and missing values (i.e., association between z_1, z_2, z_3 and x variables used to generate missing values) and only 15% missing data, the proposed multiple imputation approach selected z_1 more often than all other approaches with the exception of listwise deletion. However, delete if selected and surrogate splits outperformed the proposed multiple imputation approach in the same conditions with 30% missing data (second panel in Figure figure5). This pattern of results can be found in supplemental materials (see Table S2 in supplemental materials). When there were strong associations between the predictors and variables used to generate missing values, the proposed multiple imputation approach correctly selected z_1 more frequently than the remaining approaches, such as single imputation, delete if selected, surrogate splits, and majority rule (fourth panel in Figure figure5).

Averaging the proportion of correct first variable splits across all conditions leads to the following set of results. In the complete data conditions, z_1 was selected for the first split 98% of the time. Listwise deletion correctly identified the first split 94% of the time, which was more often than the other approaches (Table table1). The proposed multiple imputation approach correctly selected z_1 for the first variable split 88% of the time, whereas single imputation averaged 87%. Delete if selected slightly outperformed surrogate splits, but both approaches were nearly identical in correctly selecting the variable for first split 85% of the time. Majority rule selected the correct variable for the first split 56% of the time. Multiple imputation with prediction averaging did not produce a single tree structure, so this outcome was not evaluated for this approach.

3.4 Variable Importance

Variable importance values ranged from 0 to 1 for z_1, z_2, and z_3. Recall that z_1 was the target variable that contained missing values, was the first variable

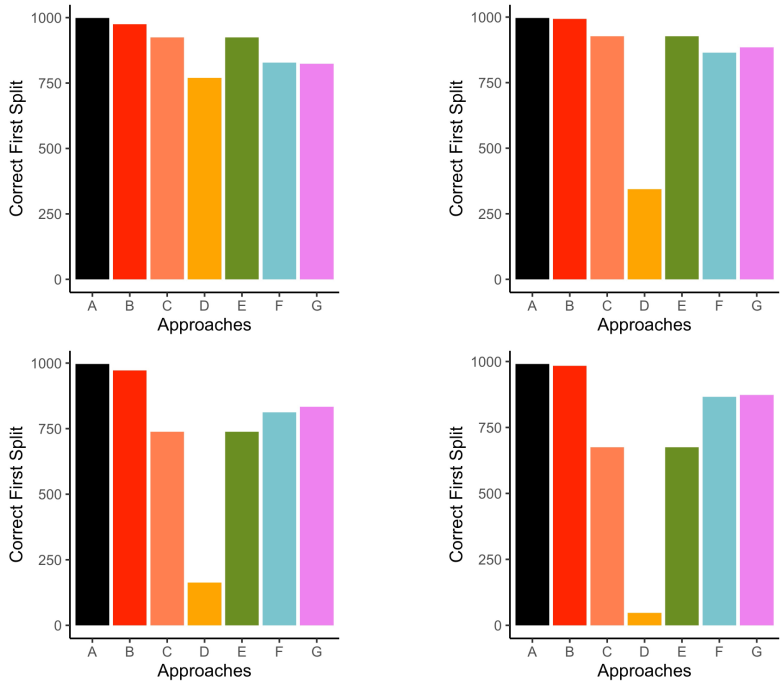


Figure 5. Correct First Variable Splits. The number of times each missing data approach correctly chose z_1 for the first split in DT out of 1,000 replications is shown in Figure figure5. Missing data approaches include: (A) Baseline - No Missing Data; (B) Listwise Deletion; (C) Delete if Selected; (D) Majority Rule; (E) Surrogate Splits; (F) Single Imputation; (G) Proposed Multiple Imputation Approach; (H) Multiple Imputation with Prediction Averaging. The first panel represents a condition where 30% of the data on z_1 were MCAR, the predictors were correlated .16, and $N = 500$. In the condition represented in the second panel (top right), 30% of the data on z_1 were MAR where a single variable predicted missing values ($R^2 = .2$), predictors were correlated .16, and $N = 500$. Third panel (bottom left) represents a condition where 30% of the data on z_1 were MAR with a single variable predicted missing values ($R^2 = .4$), predictors were correlated .36, and $N = 500$. The fourth panel represents a condition where 30% of the data on z_1 were MAR with multiple variables predicted missing values ($R^2 = .4$), predictors were correlated .36, and $N = 500$.

Table 1. Average Proportion of Correct First Variable Splits

Complete Data	Listwise Deletion	Delete if Selected	Majority Rule	Surrogate Splits	Single Imputation	MI Proposed Approach
.980	.941	.848	.562	.854	.873	.882

split, which is often associated with the greatest variable importance values. In conditions where the data were MCAR, listwise deletion most closely mimicked the variable importance values from the complete data conditions (see the left

panel of Figure figure6). Surrogate splits performed well, but tended to overestimated the importance of z_1 and z_2, and underestimated the importance of z_3, especially with larger sample sizes. The single and proposed multiple imputation approaches performed moderately well and produced nearly identical results. Both approaches underestimated the importance of z_1 and slightly overestimated the importance of the other predictors. The delete if selected and majority rule approaches mimicked the pattern for surrogate splits, but had greater discrepancy in overestimating the importance of z_1. Majority rule consistently performed poorly with respect to this outcome compared to the other missing data handling approaches.

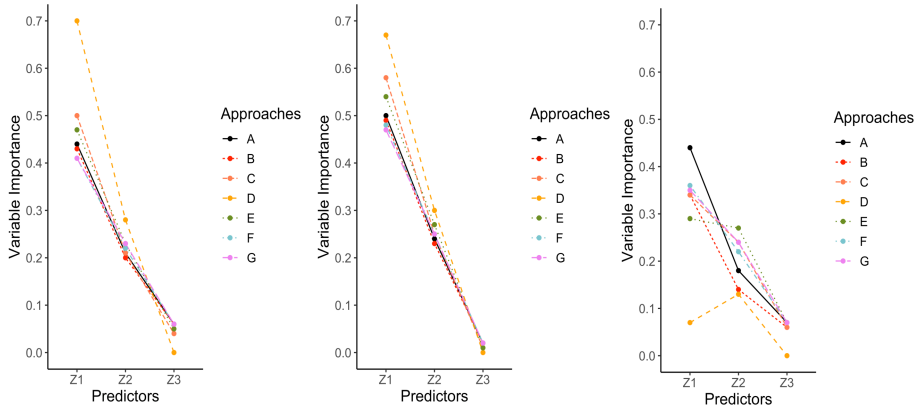


Figure 6. Variable importance measures from each missing data approach in three representative conditions. Missing data approaches include: (A) Baseline - No Missing Data; (B) Listwise Deletion; (C) Delete if Selected; (D) Majority Rule; (E) Surrogate Splits; (F) Single Imputation; (G) Proposed Multiple Imputation Approach; (H) Multiple Imputation with Prediction Averaging. The first panel represents a condition where 15% of the data on z_1 were MCAR, the predictors were correlated .36, and $N = 500$. In the condition represented in the second panel, 15% of the data on z_1 were MAR where a single variable predicted missing values ($R^2 = .2$), predictors were correlated .16, and $N = 500$. Third panel represents a condition where 30% of the data on z_1 were MAR where multiple variables predicted missing values ($R^2 = .4$), predictors were correlated .36, and $N = 200$.

The results for variable importance revealed a distinction between the MAR conditions. MAR conditions with a single variable predicting missing values and weak associations among variables had similar results when compared to MCAR conditions. However, in the more severe MAR conditions (i.e., multiple variables predicting missing values, stronger association among predictors,

high percentage of missing data), single imputation started to outperform the other approaches. Single imputation still underestimated the importance of z_1 and overestimated the other variables, but this approach had small discrepancies when compared to the complete data conditions. The proposed multiple imputation approach and delete if selected closely followed single imputation. Listwise deletion followed the same trajectory as the complete data but underestimated the importance of all predictors with larger discrepancies. Surrogate splits performed poorly in the most severe MAR conditions because it largely underestimated the importance of z_1 and overestimated the importance of z_2 (third panel in Figure figure6). Majority rule consistently had the greatest discrepancies (shown in Figure figure6).

3.5 Median Number of Splits

The median number of splits for each DT was recorded. The population tree contained six splits. The median number of splits across each approach ranged from zero to four indicating that each DT tended to underfit the data. There were little differences among most approaches across conditions. Complete data, listwise deletion, delete if selected, surrogate splits, and single imputation all had a median of two splits. The proposed multiple imputation approach often averaged one more split than the other approaches in the large sample size conditions (N = 1,000), but the overall differences were minimal. Majority rule approach averaged two splits in most conditions, but failed to find any variable to predict the outcome (i.e., resulting in zero splits) when there was a high percentage of missing values that were MAR. Multiple imputation with prediction averaging did not produce a single DT structure, so the median number of splits was not recorded.

4 Illustrative Example

Data were drawn from the Head Start Family and Child Experiences Survey 1997-2001 (FACES1997) study. The goals of FACES1997 were to (1) examine whether Head Start enhances children's development and school readiness, (2) evaluate whether Head Start strengthens families as the primary nurturers of their children, (3) determine whether Head Start provides children with high quality educational, health, and nutritional services, and (4) determine how Head Start classroom quality is related to children's outcomes. FACES1997 is a longitudinal study of 1,968 children enrolled in a Head Start program in 1997 with data collected on the cognitive, social, emotional, and physical development of Head Start children, characteristics and opinions of Head Start teachers, and characteristics and evaluations of Head Start classrooms (https://www.childandfamilydataarchive.org/cfda/archives/cfda/studies/4134).

The analytic sample contained N = 785 children who were in first grade during the 1999-2000 school year and completed cognitive testing in the spring of 2000. Of these 785 children, 370 (47%) were female. The sample was diverse

with respect to race/ethnicity. Twenty-nine percent of this subsample identified as white (non-Hispanic), 39% black (non-Hispanic), 1% Asian or Pacific Islander, and 2% Native American Indian or Alaskan. Thirty-two percent of the sample identified as Hispanic. Seventy-one percent were living below the poverty line determined by an income-to-needs ratio less than 1.0. Seventy-seven percent of families reported that at least one parent obtained a 12^{th} grade education (e.g., graduated from high school, received a GED).

These data were split into training and testing samples using a 60-40 split. Given the focus of the paper, the testing sample had complete data to make model evaluation clean, and the training data contained missing values. The training data were analyzed to develop statistical models using different missing data handling methods. DTs were overgrown and then pruned using cost-complexity pruning and k-fold cross-validation following the approach in our simulation work. Once an optimal model was determined for the training data, the model was used to generate predicted values in the testing dataset and the MSE was calculated.

The outcome variable was the Peabody Picture Vocabulary Test (PPVT; Dunnx& Dunn, 1981) standard score, which was measured in the spring of 2000. Predictor variables included a series of assessments collected during Head Start in the fall of 1997. These assessments were academic (e.g., Woodcock-Johnson Letter-Word Identification) or social (e.g., Social Skills Rating Scale) in nature.

5 Results

The DTs from each missing data handling method are shown in Figure figure7. Each terminal node contains the predicted value of the PPVT and the percent of the sample in the node. The predictor variable used to split the data is labeled within each tree node and split values are presented within the tree branches. Overall, DTs varied across methods. Note that multiple imputation with prediction averaging did not produce a consistent tree structure, so it is not included in Figure figure7. For the remaining approaches, the number of splits across DTs ranged from 1 to 13. However, many of the resulting trees shared splitting variables and values. In all remaining missing data approaches, the first splitting value was a score of 15 on identifying colors by name (COLORS). The tree produced from the surrogate split approach contained no subsequent splits. For all other approaches, the node for participants with identifying colors by name greater than or equal to 15 was split based on a value of 88 on the Woodcock-Johnson Letter-Word Identification (WJWORDSS; Woodcockx& Johnson, 1989). Notably, majority rule and single imputation had identical tree structures and did not contain any further splits. Delete if selected, listwise deletion, and the modified multiple imputation approach shared another common split value of nine on print concepts (PRCONCEPT). The DT using the modified multiple imputation approach did not contain any additional splits, whereas the listwise deletion approach contained one additional split at the value of five on McCarthy Drawing Test score (DRAWSCR; McCarthy, 1972). Lastly, the delete

if selected approach contained several additional splits beyond those described above (shown in Figure figure7).

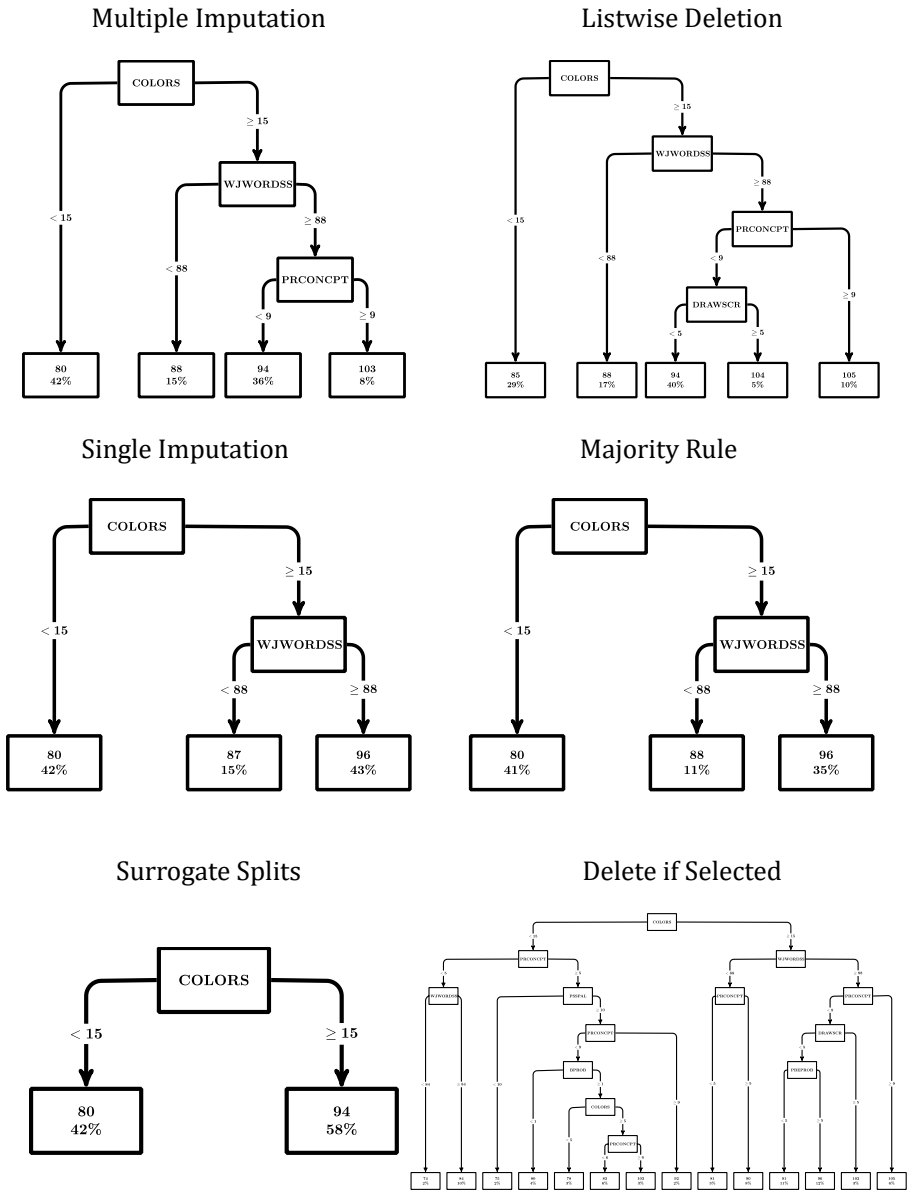


Figure 7. Illustrative Data DTs

Variable importance measures for the predictors for each DT are shown in Table table2. There was agreement across many of the missing data approaches. All missing data approaches that produced variable importance measures emphasized COLORS as an important predictor. On the other hand, a few variables were highlighted among most, but not all, missing data approaches. For example, all the approaches that were evaluated, except surrogate splits, indicated that WJWORDSS was an important variable. Nearly all approaches highlighted print concepts, The McCarthy Drawing Test score, book knowledge (BOOKKNLG), social awareness (SAWARE), and the Child Behavior Problems Index (PBEPROB; Petersonx& Zill, 1986) as important variables in all DTs, with majority rule as the exception. The last three predictors were only highlighted by a few approaches. Social skills (SSRS; Greshamx& Elliott, 1990) was deemed important with listwise deletion, delete if selected, and surrogate split approaches, whereas social skills/positive approach to learning (PSSPAL) was considered important using listwise deletion, delete if selected, and single imputation. Behavior problems total score (BPROB) was uniquely selected as an important predictor by the delete if selected approach. In summary, all approaches agreed on the variable of greatest importance (i.e., COLORS), and six out of the ten remaining predictors were highlighted in DTs using different missing data approaches.

Table 2. Illustrative Data Variable Importance

Predictors	Listwise Deletion	Delete if Selected	Majority Rule	Surrogate Splits	Single Imputation	MI Proposed Approach
COLORS	0.34	0.37	0.86	0.55	0.46	0.44
WJWORDSS	0.21	0.07	0.14	-	0.14	0.12
PRCONCPT	0.21	0.22	-	0.13	0.12	0.17
DRAWSCR	0.09	0.04	-	0.04	0.04	0.03
BOOKKNLG	0.08	0.11	-	0.11	0.10	0.11
SAWARE	0.03	0.10	-	0.14	0.13	0.12
PBEPROB	0.02	0.01	-	-	0.01	<0.01
SSRS	0.02	0.03	-	0.03	-	-
PSSPAL	0.01	0.02	-	-	<0.01	-
BPROB	-	0.02	-	-	-	-
BEARCNT	-	-	-	-	-	-

Predictions from each DT were generated for the test data. Test data contained no missing values and consisted of 314 participants. To evaluate prediction accuracy, the MSE (i.e., average squared difference of estimated scores from DTs and actual scores on test data) was calculated for each missing data approach (see Table table3). Overall, listwise deletion produced the best prediction of PPVT in the test data. The proposed multiple imputation approach had the second-best performance. Majority rule, single imputation, and multiple imputation with prediction averaging performed similarly to the proposed multiple imputation

approach with only minor increases in MSE. Delete if selected performed poorly, and surrogate splits had the poorest performance.

Table 3. Illustrative Data MSE and R^2

Measures	Listwise Deletion	Delete if Selected	Majority Rule	Surrogate Splits	SI	MI Proposed Approach	MI Prediction Averaging
MSE	134.60	149.85	145.16	156.24	146.07	144.00	146.17
R^2	0.37	0.35	0.36	0.25	0.36	0.37	0.33

Note. SI: Single Imputation.

We also calculated an R^2 value to measure predictive quality of each missing data approach (shown in Table table3). Specifically, R^2 was calculated as the squared correlation between the predicted and observed outcome values using the test data. It represents the percent of variance in test data PPVT scores accounted for by the prediction model using each missing data approach. Thirty-seven percent of the variance in PPVT scores was accounted for by the predicted values produced by the listwise deletion approach. Similarly, 37% of the variance in PPVT scores was accounted for by predicted scores from the modified multiple imputation approach. Single imputation and majority rule approach led to R^2 values of 36% and delete if selected led to an R^2 of 35%. Multiple imputation with prediction averaging had an R^2 of 33% and the DT using surrogate splits had an R^2 of 25%. In summary, listwise deletion and the modified multiple imputation approach led to DTs that performed best in the test dataset.

6 Discussion

A modified multiple imputation approach was proposed for handling missing data in DTs. The proposed approach involves four steps: (1) Impute missing values, (2) Fit a DT to the imputed dataset, prune the DT using k-fold cross validation, and retain the associated *cp* value, (3) Repeat steps 1 and 2 multiple times, and (4) stack all imputed datasets into a single data frame, fit a DT to the stacked dataset, and using the averaged *cp* value from when the DTs were fit to each imputed dataset. A simulation was conducted to compare the proposed approach to listwise deletion, delete if selected, majority rule, surrogate splits, single imputation, and multiple imputation with prediction averaging under multiple MAR and MCAR conditions.

6.1 Summary of Findings

Overall, all missing data approaches produced DTs with better performance in conditions with larger sample sizes and lower rates of missing values. Across the outcome measures, the proposed multiple imputation method performed better than the other approaches when data were MAR with a strong association

between multiple predictors and missing values. Additionally, the proposed multiple imputation approach handled small sample sizes ($N = 200$) better than the other approaches across the MAR conditions. On the other hand, surrogate splits performed the best when data were MCAR and when data were MAR with a single predictor that had a weak association with missing values. It appears the weak associations in these MAR conditions led to conditions that were close to MCAR.

In addition to the simulation work, empirical data from FACES 1997-2001 were analyzed to compare the seven approaches for handling missing data. A series of assessments were taken on a total of $N = 785$ children. We found that listwise deletion and multiple imputation had the highest prediction accuracy as measured by MSE and R^2. Majority rule, single imputation, and delete if selected had relatively high prediction accuracy. Surprisingly, multiple imputation with prediction averaging had lower prediction accuracy and surrogate splits had the worst prediction accuracy.

6.2 Recommendations

The results of our simulation research leads to the following set of recommendations. The proposed multiple imputation approach is recommended in situations where data are MAR, especially when dealing with small sample sizes. Surrogate splits are recommended when data are MCAR or mildly MAR (i.e., data are MAR with weak associations and a fairly large sample sizes, $N \geq 500$). If a researcher is only interested in prediction accuracy and has no interest in interpreting the DT, multiple imputation with prediction averaging is recommended for either MAR and MCAR data. However, in these situations, an ensemble method, such as random forests (Breiman, 2001) or boosting (Breiman, 1998; Friedman, 2002), may be preferred. Single imputation is a simple approach, but is not recommended over the proposed multiple imputation approach because it often underperformed by comparison.

Listwise deletion, delete if selected, and majority rule are not generally recommended. Both listwise deletion and delete if selected could be recommended when data are MCAR and there is a small percentage of missing data. Deletion approaches may be a relatively simple and convenient method for handling missing data in such situations, but these methods proved inferior in most conditions. Majority rule generally had the poorest performance across all conditions and is not recommended.

6.3 Limitations and Future Directions

A limitation of this study is that missing data were handled with a single type of imputation. A variety of imputation methods have been developed in statistical frameworks, which are typically built upon linear or logistic regression models. However, imputation models have also been built upon partitioning algorithms, such as DTs and random forests (Tangx& Ishwaran, 2017), and these imputation approaches were not considered.

A second limitation is that we only considered one pooling approach in the proposed multiple imputation approach. That is, when analyzing the stacked multiply imputed dataset, the *cp* was set to the average value obtained from analyzing each imputed dataset. Another metric may be more appropriate instead of the average. For example, the minimum value of the *cp* or the 5^{th} percentile would lead to larger DTs and may be more appropriate because the resulting DTs were smaller than the population DT. More research is needed to determine the optimal approach to determining the size of the DT with the stacked multiply imputed data.

Another consideration is that this study evaluated how well missing data approaches performed when the predictors contain missing values and the outcome variable does not. The nature of the missing values was manipulated only on the first splitting variable, z_1. However, in practice, missing values may appear across both the predictors and outcome variable. Future studies should consider how to treat the case where values on the outcome variable are missing.

6.4 Concluding Remarks

The proposed modified multiple imputation approach for handling missing data in DTs was found to outperform surrogate splits, the default approach in several DT packages, for handling MAR data, particularly in small samples. To our knowledge, multiple imputation has only been implemented in DTs by averaging predicted values from different tree structures fit to each imputed dataset (Feelders, 1999; Twala, 2009). Our proposed modified multiple imputation approach leads to a single DT so that a single set of splitting variables can be interpreted.

Machine learning techniques are becoming more widely accepted in the social and behavioral sciences where missing data are a common problem. Additional research is needed to more fully examine how different machine learning algorithms, including different DT algorithms, such as conditional inference trees (Hothorn, Hornik,x& Zeileis, 2006) and evolutionary trees (De Jong, 2006; Eiben, 2003; Fogel, Bäck,x& Michalewicz, 2000), perform under a variety of missing data conditions and whether novel missing data approaches can improve upon the default strategies. We look forward to this research.

References

Agresti, A. (2012). *Categorical data analysis* (3rd ed.). Wiley.

Allison, P. (2002). *Missing data.* SAGE Publications, Inc.

Baraldi, A. N., & Enders, C. K. (2010). An introduction to modern missing data analyses. *Journal of School Psychology*, *48*(1), 5-37. doi: https://doi.org/10.1016/j.jsp.2009.10.001

Batista, G. E. A. P. A., & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, *17*(5-6), 519-533. doi: https://doi.org/10.1080/713827181

Beaulac, C., & Rosenthal, J. S. (2020). Best: A decision tree algorithm that handles missing values. *Computational Statistics*, *35*(3), 1001–1026. doi: https://doi.org/10.1007/s00180-020-00987-z

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123-140. doi: https://doi.org/10.1007/bf00058655

Breiman, L. (1998). Arcing classifiers. *The Annals of Statistics*, *26*(3), 801–824.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5-32. doi: https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Taylor & Francis.

De Jong, K. A. (2006). *Evolutionary computation: A unified approach*. MIT Press.

Dunn, L. M., & Dunn, L. M. (1981). *Peabody picture vocabulary test-revised*. American Guidance Service, Inc.

Eiben, A. E. (2003). *Introduction to evolutionary computing*. Springer.

Enders, C. K. (2003). Using the expectation maximization algorithm to estimate coefficient alpha for scales with item-level missing data. *Psychological Methods*, *8*(3), 322–337. doi: https://doi.org/10.1037/1082-989x.8.3.322

Enders, C. K. (2010). *Applied missing data analysis*. Guilford Press.

Enders, C. K., Dietz, S., Montague, M., & Dixon, J. (2006). Applications of research methodology. In T. E. Scruggs & M. A. Mastropieri (Eds.), (Vol. 19, pp. 101–129). Emerald Group Publishing Limited.

Feelders, A. (1999). Principles of data mining and knowledge discovery. In J. M. Żytkow & J. Rauch (Eds.), (pp. 329–334). Springer.

Fogel, D. B., Bäck, T., & Michalewicz, Z. (2000). *Evolutionary computation*. Institute of Physics Publishing.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis,*, *38*(4), 367–378. doi: https://doi.org/10.1016/s0167-9473(01)00065-2

Gonzalez, O., O'Rourke, H. P., Wurpts, I. C., & Grimm, K. J. (2018). Analyzing monte carlo simulation studies with classification and regression trees. *Structural Equation Modeling: A Multidisciplinary Journal*, *25*(3), 403-413. doi: https://doi.org/10.1080/10705511.2017.1369353

Gresham, F. M., & Elliott, S. N. (1990). *Social skills rating system manual*. American Guidance Service.

Groves, R. M., & Peytcheva, E. (2008). The impact of nonresponse rates on nonresponse bias a meta-analysis. *Public Opinion Quarterly*, *72*(2), 167-189. doi: https://doi.org/10.1093/poq/nfn011

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

Hattie, J. (1983). The tendency to omit items: Another deviant response characteristic. *Educational and Psychological Measurement*, *43*(4), 1041–1045. doi: https://doi.org/10.1177/001316448304300412

Holman, R., & Glas, C. A. W. (2005). Modelling non-ignorable missing-data mechanisms with item response theory models. *British Jour-*

nal of Mathematical and Statistical Psychology, *58*(1), 1-17. doi: https://doi.org/10.1111/j.2044-8317.2005.tb00312.x

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651-674. doi: https://doi.org/10.1198/106186006x133933

Huggins-Manley, A. C., Algina, J., & Zhou, S. (2018). Models for semiordered data to address not applicable responses in scale measurement. *Structural Equation Modeling: A Multidisciplinary Journal*, *25*(2), 230-243. doi: https://doi.org/10.1080/10705511.2017.1376586

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in r.* Springer-Verlag.

Johnson, V. E., & Albert, J. H. (1999). *Ordinal data modeling.* Springer-Verlag.

Lavrakas, P. (2008). *Encyclopedia of survey research methods.* SAGE Publications, Inc.

Little, R. J. A., & Rubin, D. B. (2002). Statistical analysis with missing data. In (pp. 59–74). John Wiley & Sons, Ltd.

Loh, W. Y. (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, *1*(1), 14-23. doi: https://doi.org/10.1002/widm.8

Mazza, G. L., Enders, C. K., & Ruehlman, L. S. (2015). Addressing item-level missing data: A comparison of proration and full information maximum likelihood estimation. *Multivariate Behavioral Research*, *50*(5), 504-519. doi: https://doi.org/10.1080/00273171.2015.1068157

Moustaki, I., & Knott, M. (2000). Weighting for item non-response in attitude scales by using latent variable models with covariates. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *163*(3), 445-459. doi: https://doi.org/10.1111/1467-985x.00177

Muthén, B., Kaplan, D., & Hollis, M. (1987). On structural equation modeling with data that are not missing completely at random. *Psychometrika*, *52*(3), 431-462. doi: https://doi.org/10.1007/bf02294365

Peterson, J. L., & Zill, N. (1986). Marital disruption, parent–child relationships, and behavior problems in children. *Journal of Marriage and the Family*, *48*(2), 295-307. doi: https://doi.org/10.2307/352397

R Core Team. (2020). R: A language and environment for statistical computing. [Computer software manual]. Vienna, Austria..

Raghunathan, T. E. (2004). What do we do with missing data? some options for analysis of incomplete data. *Annual Review of Public Health*, *25*(1), 99-117. doi: https://doi.org/10.1146/annurev.publhealth.25.102802.124410

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581-592. doi: https://doi.org/10.1093/biomet/63.3.581

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys.* John Wiley & Sons Inc.

Schafer, J. L., & Olsen, M. K. (1998). Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate Behavioral Research*, *33*(4), 545-571. doi:

https://doi.org/10.1207/s15327906mbr3304_5

Tang, F., & Ishwaran, H. (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining*, *10*(6), 363-377. doi: https://doi.org/10.1002/sam.11348

Therneau, T., & Atkinson, B. (2019). rpart: Recursive partitioning and regression trees (4.1-15) [Computer software manual]. https://CRAN.R-project.org/package=rpart.

Twala, B. (2009). An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence*, *23*(5), 373-405. doi: https://doi.org/10.1080/08839510902872223

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, *45*(1), 1-67. doi: https://doi.org/10.18637/jss.v045.i03

Woodcock, R. W., & Johnson, M. B. (1989). *Woodcock-johnson tests of achievement.* Riverside Publishing.

Journal of Behavioral Data Science, 2021, *1* (1), 154–169.
DOI: https://doi.org/10.35566/jbds/v1n1/p8

COVID-19 Outbreak Prediction and Analysis using Self Reported Symptoms

Rohan Sukumaran*[1], Parth Patwa*[1], Sethuraman T V*[1], Sheshank Shankar[1], Rishank Kanaparti[1], Joseph Bae[1,2], Yash Mathur[1], Abhishek Singh[4], Ayush Chopra[4], Myungsun Kang[1], Priya Ramaswamy[1,3], and Ramesh Raskar[1,4]

[1] PathCheck Foundation
rohan.sukumaran@pathcheck.org, parth.patwa@pathcheck.org
sethu.ramantv@pathcheck.org
[2] Stony Brook Medicine
[3] University of California San Francisco
[4] MIT Media Lab

Abstract. It is crucial for policymakers to understand the community prevalence of COVID-19 so combative resources can be effectively allocated and prioritized during the COVID-19 pandemic. Traditionally, community prevalence has been assessed through diagnostic and antibody testing data. However, despite the increasing availability of COVID-19 testing, the required level has not been met in parts of the globe, introducing a need for an alternative method for communities to determine disease prevalence. This is further complicated by the observation that COVID-19 prevalence and spread vary across different spatial, temporal, and demographic verticals. In this study, we study trends in the spread of COVID-19 by utilizing the results of self-reported COVID-19 symptoms surveys as a complement to COVID-19 testing reports. This allows us to assess community disease prevalence, even in areas with low COVID-19 testing ability. Using individually reported symptom data from various populations, our method predicts the likely percentage of the population that tested positive for COVID-19. We achieved a mean absolute error (MAE) of 1.14 and mean relative error (MRE) of 60.40% with 95% confidence interval as [60.12, 60.67]. This implies that our model predicts +/- 1140 cases than the original in a population of 1 million. In addition, we forecast the location-wise percentage of the population testing positive for the next 30 days using self-reported symptoms data from previous days. The MAE for this method is as low as 0.15 (MRE of 11.28% with 95% confidence interval [10.9, 11.6]) for New York. We present an analysis of these results, exposing various clinical attributes of interest across different demographics. Lastly, we qualitatively analyze how various policy enactments (testing, curfew) affect the prevalence of COVID-19 in a community.

* Equal contribution.

Keywords: Machine Learning · COVID-19 · Outbreak Prediction · Time Series

1 Introduction

The rapid progression of the COVID-19 pandemic has provoked large-scale data collection efforts on an international level to study the epidemiology of the virus and inform policies. Various studies have been undertaken to predict the spread, severity, and unique characteristics of the COVID-19 infection, across a broad range of clinical, imaging, and population-level datasets (Gostic, Gomez, Mummah, Kucharski, & Lloyd-Smith, 2020; Liang et al., 2020; Menni et al., 2020; Shi et al., 2020). For instance, Menni et al. (2020) use self-reported data from a mobile app to predict a positive COVID-19 test result based upon symptom presentation. Anosmia was shown to be the strongest predictor of disease presence, and a model for disease detection using symptoms-based predictors was indicated to have a sensitivity of about 65%. Studies like Parma et al.(2020) have shown that ageusia and anosmia are widespread sequelae of COVID-19 pathogenesis. From the onset of COVID-19, there also has been a significant amount of work in mathematical modeling to understand the outbreak under different situations for different demographics (Menni et al., 2020; Saad-Roy et al., 2020; Wilder, Mina, & Tambe, 2020). However, these works primarily focus on the population level. Further, the estimation of different transition probabilities to move between compartments is challenging.

Carnegie Mellon University (CMU) and the University of Maryland (UMD) have built chronologically aggregated datasets of self-reported COVID-19 symptoms by conducting surveys at national and international levels (Delphi group, 2020; Fan et al., 2020). The surveys contain questions regarding whether the respondent has experienced several of the common symptoms of COVID-19 (e.g. anosmia, ageusia, cough, etc.) in addition to various behavioral questions concerning the number of trips a respondent has taken outdoors and whether they have received a COVID-19 test.

In this work, we perform several studies using the CMU (Delphi group, 2020), UMD (Fan et al., 2020), and OxCGRT (Hale, Webster, Petherick, Phillips, & Kira, 2020) datasets. Our experiments examine correlations among variables in the CMU data to determine which symptoms and behaviors are most correlated to high percentages of Covid Like Illness (CLI). We investigate how the different symptoms impact the percentage of populations with CLI across different spatio-temporal and demographic (age, gender) settings. We also predict the percentage of population who got tested positive for COVID-19 and achieve 60% Mean Relative Error. Further, our experiments involve time-series analysis of these datasets to forecast CLI over time. Here, we identify how different spatial window trends vary across different temporal windows. We aim to use the findings from this method to understand the possibilities of modeling CLI for geographic areas in which data collection is sparse or non-existent. Furthermore, results from our experiments can potentially guide public health policies

for COVID-19. Understanding how the disease is progressing can help the policymakers introduce non-pharmaceutical interventions (NPIs) and also help them understand how to distribute critical resources (medicines, doctors, healthcare workers, transportation, and more). This could now be done based on the insights provided by our models, instead of relying completely on clinical testing data. Prediction of outbreaks using self-reported symptoms can also help reduce the load on testing resources. Similar self reported data and survey data have been used by (Rodriguez, Muralidhar, et al., 2020; Rodriguez, Tabassum, et al., 2020; Garcia-Agundez et al., 2021) for understanding the pandemic and drawing actionable insights.

2 Datasets

The **CMU Symptom Survey** aggregates the results of a survey run by CMU (Delphi group, 2020) that was distributed across the US to approx 70k random Facebook users daily. It gives a set of indicators that can inform our reasoning about the pandemic. The indicators include:

- Symptoms related indicators like the percentage of respondents reporting fever and the percentage of respondents reporting sore throat.
- Pre-existing medical condition related indicators like the percentage of respondents having diabetes and the percentage of respondents having Autoimmune Disorder.
- Behavior related indicators like the percentage of respondents who avoid contact with others most of the time and the percentage of respondents who worked outside home.

The data set has a total of 104 columns (in October 2020), including weighted (adjusted for sampling bias), unweighted signals, and demographic information (age, gender, etc.) at county and state level. In this study, we use the state level data from Apr. 4, 2020 to Sep. 11, 2020, which is henceforth referred to as the CMU dataset in the paper.

The **UMD Global Symptom Survey** aggregates the results of a survey conducted by UMD through Facebook (Fan et al., 2020). The survey is available in 56 languages. A representative sample of Facebook users were invited on a daily basis to report on topics including symptoms and social distancing behavior. Facebook provides weights to reduce non-response and coverage bias. Country and region-level statistics are published daily via the public API and dashboards, and micro-data is available for researchers via data use agreements. Over half a million responses were collected daily. We use the data of 968 regions, available from May 1 to September 11, 2020. There are 49 (in October 2020) unweighted signals, as well as their weighted forms (adjusted for sampling bias).

The **Oxford COVID-19 Government Response Tracker (OxCGRT)** (Hale et al., 2020) contains government COVID-19 policy data as a numerical scale value representing the extent of government action. OxCGRT collects publicly available information on 20 indicators of government response. This

information was collected by a team of over 200 volunteers from the Oxford community and was updated continuously. The data set also includes statistics on the number of reported Covid-19 cases and deaths in each country, which were taken from the JHU CSSE (Dong, Du, & Gardner, 2020) data repository for all countries and the US.

The **Prevalence of Self-Reported Obesity by State and Territory, BRFSS, 2019 - CDC** (CDC, 2020) is a dataset published by CDC containing the aggregated self-reported obesity values. The data are at the state level and contain the obesity values and confidence intervals (95%). This dataset contains other information like race, ethnicity, and food habits that can be used for further analysis.

3 Methods and Experiments

Different methods and strategies have been used to analyze the data. Our code used in the analysis is publicly available at `https://github.com/PrivateKit/CovidSymptomChallenge`.

3.1 Correlation Studies

Correlations between features of the datasets provide crucial information about the features and the degree of influence they have over the target value. We conducted correlation analysis on different subgroups like symptomatic and asymptomatic subjects, and varying demographic regions in the CMU dataset to discover relationships among the signals and with the target variable. We also investigated the significance of obesity and population density on the susceptibility to COVID-19 at the state level (CDC, 2020). Refer to the supplementary materials for more information.

3.2 Feature Pruning

We first dropped demographic features such as date, gender, and age. Next, we dropped the unweighted features because their weighted counterparts were used. We also dropped features including the percentage of people who tested negative, the weighted percentage of people who tested positive because they were directly related to testing and would make the prediction trivial. Furthermore, we dropped the derived features such as the estimated percentage of people with influenza-like illness because they were not directly reported by the respondents. Finally, we dropped some features with aggregated information such as the average number of people in respondent's household who have Covid Like Illness. After the entire process, we selected 36 features. The selected feature list is provided in the supplementary materials.

3.3 Outbreak Prediction

We predicted the percentage of the population that tested positive at the state level from the CMU dataset. We ranked these 36 signals using *f_regression* ("sklearn f regression", 2007-2020) (*f_statistic* of the correlation to the target variable) and predicted the target variable using the top n ranked features. We experimented with the *top n features* value from 1 to 36 for various demographic groups. We trained linear regression (Galton, 1886), decision tree (Quinlan, 1986), and gradient boosting (Friedman, 2001) models. All the models were implemented using scikit-learn (Pedregosa et al., 2011). We used 80% of the data for training and the remaining 20% of the data for testing. The data were split randomly.

3.4 Time Series Analysis

We predicted the percentage of people that tested positive using the CMU dataset and the percentage of people with CLI with the UMD dataset. We independently used the top "n" features (according to their ranking obtained in outbreak prediction and empirical evidence combined with human experts ranking) from the CMU (36) and UMD (49) datasets for multivariate multi-step time series forecasting. Given the data spread across different spatial windows (geographies) at the state level, we employed an agglomerative clustering method independently on symptoms and behavioral/external patterns, and sample locations that were not in the same cluster for our analysis. Using the Augmented Dickey-Fuller test (Cheung & Lai, 1995), we found the time series samples for these spatial windows to be stationary. Furthermore, we bucketed the data based on the age and gender of the respondents, to provide granular insights on the model performance on various demographics. With a total of 12 demographic buckets [(age, gender) pairs], we used a vector autoregressive (VAR) (Holden, 1995) model and an LSTM (Gers, Schmidhuber, & Cummins, 1999) model for the experiments. Furthermore, we qualitatively evaluated the impact of government policies, e.g., curfew, on the spread of the virus. We used 80% of the data for training and the remaining 20% of the data for testing.

4 Results and Discussion

4.1 Correlation Studies

The state level analysis revealed a moderate positive correlation, $r = 0.24$ (p-value < 0.001), between the percentage of people tested positive and the statewide obesity level. Here, the obesity is defined as BMI$>$ 30.0 (NIH, 2020). The results are consistent with prior clinical studies like (Chan et al., 2020) and indicate that further research is required to investigate if the lack of certain nutrients like Vitamin B, Zinc, Iron, or having a BMI$>$ 30.0 could make an individual more susceptible to COVID-19. Figure 1 shows the correlations among multiple self-reported symptoms and the symptoms with significant positive correlations

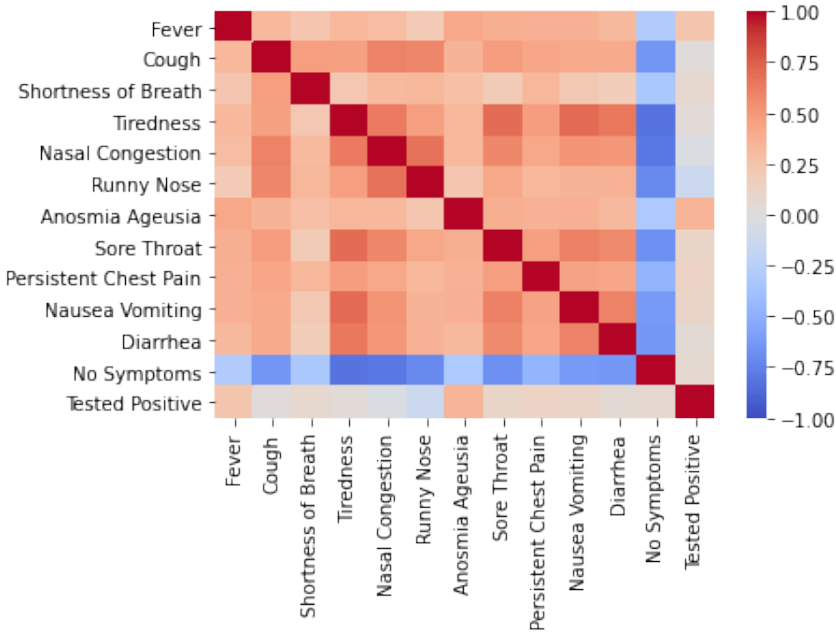


Figure 1. Correlation among self-reported symptoms and the percent of population tested COVID positive.

are highlighted. This clearly reveals that anosmia, ageusia and fever are relatively strong indicators of COVID-19. From Figure 2, we see that contact with a COVID-19 positive individual is strongly correlated with testing COVID-19 positive. Conversely, the percentage of population who avoid outside contact and the percentage of population testing positive for COVID-19 have a negative correlation. We also found a moderate positive correlation between the population density and the percentage of population reporting positive COVID-19, which indicates easier transmission of the virus in a congested environment. These observations reaffirm the highly contagious nature of the virus and the need for social distancing.

The results motivated us to estimate the percentage of people who tested COVID-19 positive based on the percentage of people who had a direct contact with anyone who recently tested positive. In doing so, we achieve a mean relative error (MRE) of 2.33% and a mean absolute error (MAE) of 0.03.

Here, MAE is the absolute value of the difference between the predicted value and the actual value, averaged over all data points:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i|,$$

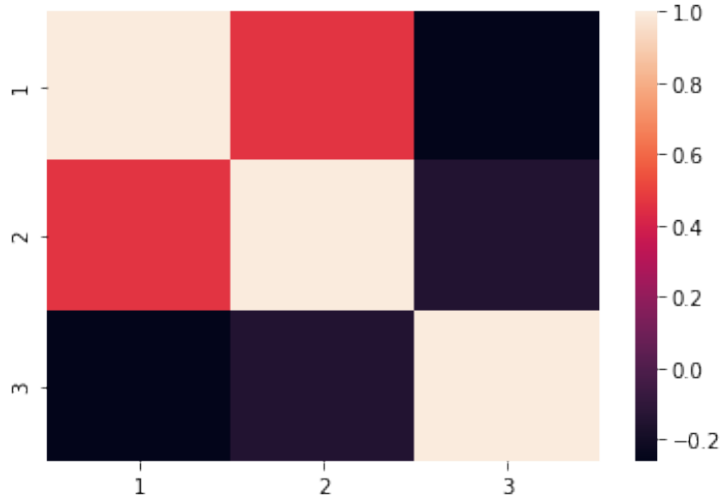


Figure 2. Correlations between the percent of people having contact with someone having CLI and the percent of people who tested positive. Here, the attribute (1) = percentage of people who had contact with someone having COVID-19, (2) = percentage of people tested positive, (3) = percentage of people who avoided contact all/most of the time.

where n is the total data instances, y_i is the predicted value and x_i is the actual value. Relative error is the absolute difference between the predicted value and the actual value, divided by the actual value. MAE is the relative error averaged over all the data points:

$$\text{MRE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - x_i}{x_i + 1}\right|,$$

where 1 is added in the denominator to avoid division by 0.

We found that a low MAE value can be misleading in the case of predicting the spread of the virus. The MAE for the outbreak prediction was low and had a small range (1-1.4) but more than 75% of the target lied between 0-2.6, meaning only a small percentage of the entire population had COVID-19 (if 1% of the entire population was affected and an MAE of 1 indicates the predicted cases could double the actual cases). MRE accounts for even minute changes (errors) in the prediction. Hence, it is a better metric to judge a system.

4.2 Policies vs CLI/Community Sick Impacts

The impacts of different non-pharmaceutical interventions (NPIs) could be analyzed by combining the CMU, UMD, and Oxford data. A particular analysis

from that is reported here, where we noticed that lifting of stay at home restrictions resulted in a sudden spike in the number of cases. This is visualized in figure 3.

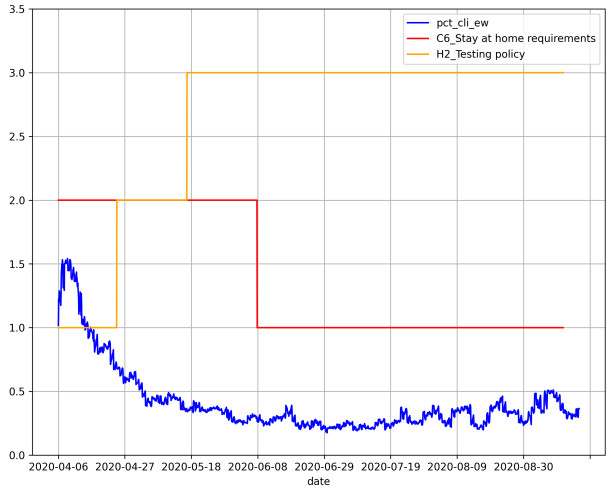


Figure 3. Policy impacts: When Stay at home restrictions were stronger, even with higher testing rates, the percentage of population with CLI (pct_cli_ew) had a downward trend.

4.3 Outbreak prediction on CMU Dataset

Gradient boosting performed the best and considerably better than the next best algorithm in terms of the error metrics for every demographic group. Hence, only the results for Gradient Boosting are presented. Table 1 shows the best accuracy achieved per dataset. For every dataset, the best "n" number of features is about 30. We achieved an MRE of 60.40% for the entire dataset. The performance was better on the female-only data when compared to the male-only data. The performance was slightly better on 55+ age data than other age groups. This can also be observed from figure 4.

4.3.1 Top Features Except for minor reordering, the top 5 features were CLI in community, loss of smell, CLI in house hold (HH), fever in HH, and fever across every data split. The top 6 to 10 features per data split are given in figure 5. We can see that 'worked outside home' and 'avoid contact most time'

Table 1. Results of gradient boosting model for the prediction of the percentage of population tested positive across demographics. The mean relative error (MRE) and mean absolute error (MAE) are average of 20 runs. The 95% confidence interval (CI) for MRE is calculated on 20 runs (data were shuffled randomly each time).

Demographic	best n	MAE	MRE	CI
Entire	35	1.14	60.40	(60.12, 60.67)
Male	34	1.38	78.14	(77.67, 78.62)
Female	36	1.10	56.89	(56.48, 57.30)
Age 18-34	30	1.23	66.35	(65.59, 67.12)
Age 35-54	35	1.29	67.59	(67.13, 68.04)
Age 55+	33	1.20	66.40	(65.86, 66.94)

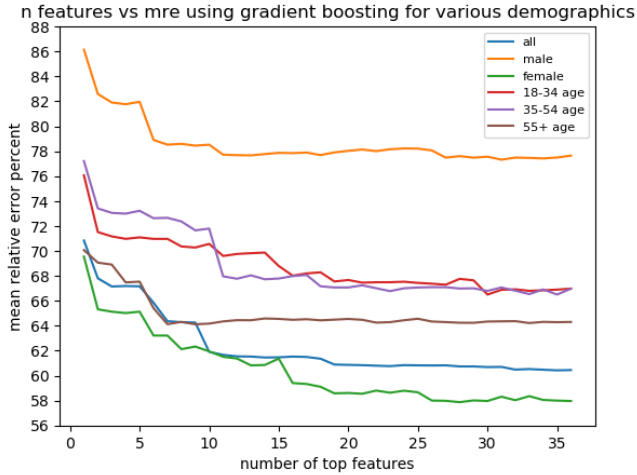


Figure 4. Error vs. the number of top features used for the gradient boosting model. Errors vary across demographics and generally decrease with the increase of the number of features (n). The decrease is not considerable after n = 20.

are useful features for male, female, and 55+ age groups. Figure 4 shows MRE vs. the number of features selected for different data splits. Overall, the error decreased as we added more features. However, the decrease in error was not considerable when we went beyond 20 features ($< 1\%$).

4.4 Time Series Analysis

As seen in Tables 2 and 3, we were able to forecast the PCT_CLI with an MRE of 15.31% using just 23 features from the UMD dataset for Lombardia and with an MRE of 42.72% for Northern Ireland. The 23 features (provided in the supplementary materials) were selected with the help of human experts and empirical analysis. We can see that VAR performed better than LSTM on average. This can be explained by the dearth of data available. Furthermore, we can see that

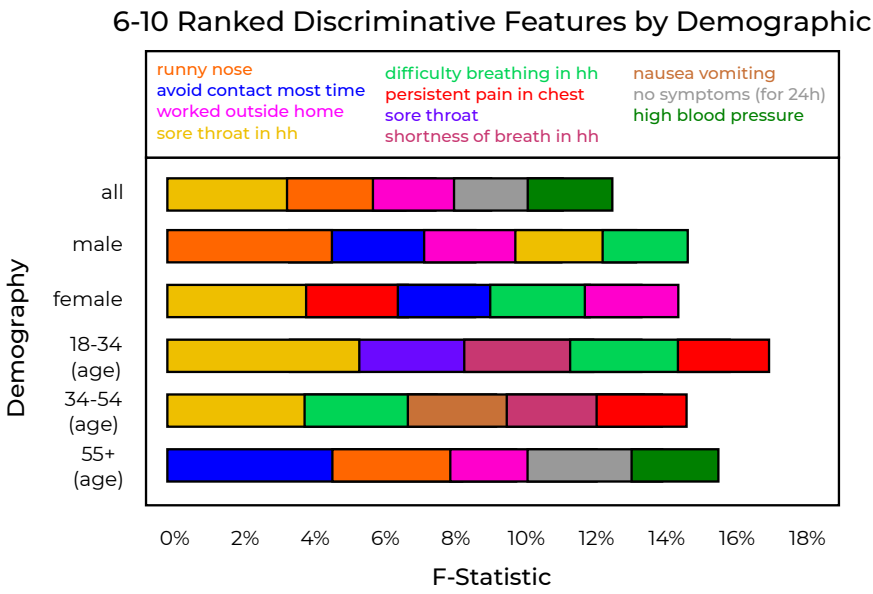


Figure 5. After the top 5 predictive features (which were roughly identical), there are considerable differences between the most predictive features segmented across demographics. For example, for the age 34-55 group, 'sore throat in hh (household)' was the sixth most predictive feature but it is not even in the top 10 most predictive features for the 55+ age group.

the outbreak forecasting for New York achieved 11.28% MRE, making use of only 10 features (these features were selected based on the outbreak prediction results and further empirically identified as well). This might be caused by an inherent bias in the sampling strategy or participant responses. For example, the high correlation noted between anosmia and COVID-19 prevalence suggested several probable causes of confounding relationships between the two. This could also occur if both symptoms were specific and sensitive for COVID-19 infection.

4.5 Symptoms vs CLI overlap

The percentage of population with symptoms like cough, fever, and runny nose was much higher than the percentage of people who suffered from CLI or the percentage of people who were sick in the community. Only 4% of the people in the UMD dataset who reported having CLI did not suffer from chest pain and nausea.

Table 2. The errors of forecasting the outbreak of COVID-19 (the percentage of people who tested positive) for the next 30 days using VAR and LSTM.

Location	*MRE*	*MAE*
	VAR	
New York	11.28, 95% CI [10.9, 11.6]	0.15
California	13.48, 95% CI [13.4, 13.5]	0.23
Florida	17.49, 95% CI [17.5, 17.5]	0.38
New Jersey	17.93, 95% CI [17.9, 18]	0.26
	LSTM	
New York	23.61, 95% CI [23.6, 23.7]	0.36
California	45.06, 95% CI [45, 45.2]	0.91
Florida	64.98, 95% CI [64.8, 65.1]	1.51
New Jersey	15.78, 95% CI [15.7, 15.9]	0.26

Table 3. Results of forecasting the outbreak of COVID-19 (the percentage of people with COVID-19 like illness in the population - PCT_CLI) for the next 30 days using the VAR and LSTM models.

Location	*MRE*	*MAE*
	VAR	
Tokyo	17.77, 95% CI [17.7, 17.8]	0.28
British Columbia	21.35, 95% CI [21.3, 21.4]	0.34
Northern Ireland	42.72, 95% CI [42.7, 42.8]	0.87
Lombardia	15.31, 95% CI [15.3, 15.4]	0.22
	LSTM	
Tokyo	30.00, 95% CI [29.9, 30.1]	0.53
British Columbia	31.11, 95% CI [30.9, 31.3]	0.56
Northern Ireland	42.46, 95% CI [42.1, 42.9]	1.21
Lombardia	16.11, 95% CI [16, 16.2]	0.21

4.6 Ablation Studies

We performed ablation studies to verify and investigate the relative importance of the features that were selected using f_regression feature ranking algorithm ("sklearn f regression", 2007-2020). In the following experiments, the top $N = 10$ features obtained from the f_regression algorithm are considered as the subset for evaluation.

4.6.1 All-but-one experiment In this experiment, the target variable which is the percentage of people affected by COVID-19 was estimated by considering $N - 1$ features from a given set of top N features by dropping 1 feature at a time in every iteration in descending order. The results were visualized in figure 6 from which it is clear that there was a considerable increased error when the most significant feature was dropped and the loss in performance was not as drastic when any other feature was dropped. This reaffirms our feature selection method.

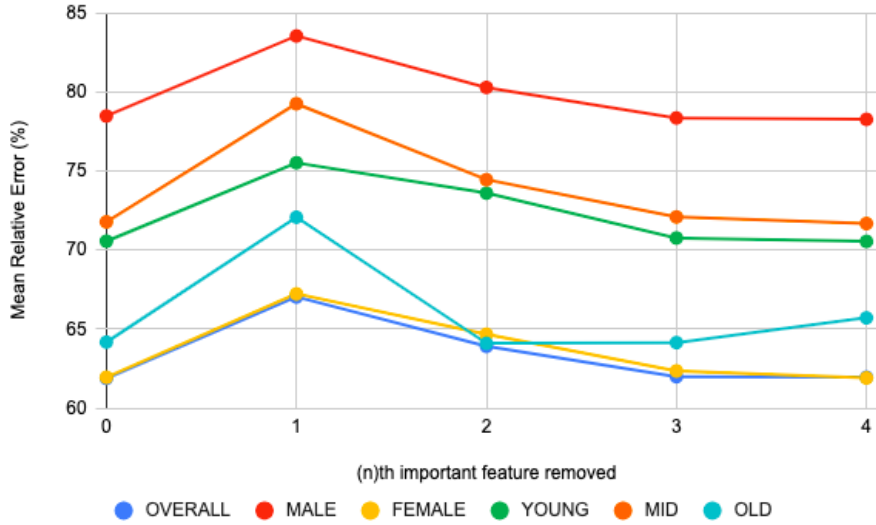


Figure 6. Results of all-but-one experiment (MRE).

4.6.2 Cumulative Feature Dropping In this experiment, we estimated the target variable based on the top N=10 features and then carried out the experiment with $N-i$ features in every iteration where i was the iteration count. The features were dropped in descending order. Figure 7 shows the results. The change in slope from the start to the end of the graph shows that the most important feature had a huge significance of the performance. This observation reinforces the inference of the all-but-one experiment and validates our feature selection algorithm.

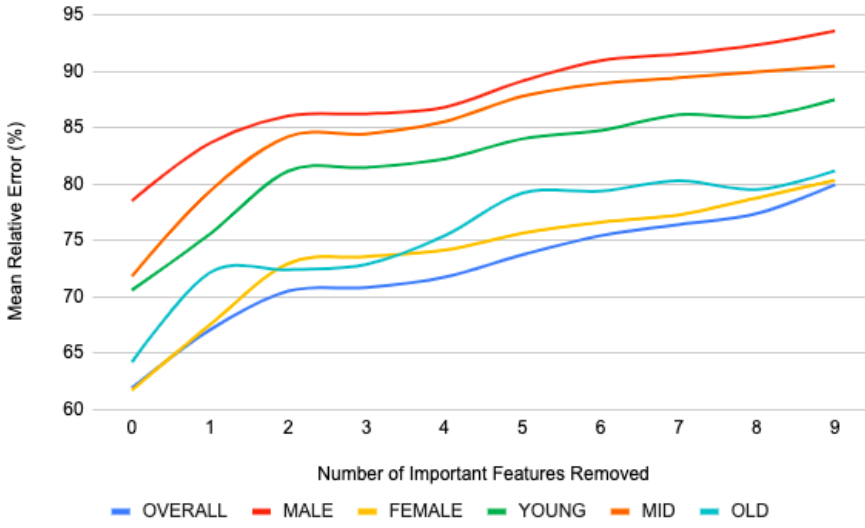


Figure 7. Results of cumulative feature dropping.

5 Conclusion And Future Work

In this work, we analyzed the benefits of the COVID-19 self-reported symptoms presented in the CMU, UMD, and Oxford datasets. We conducted correlation analysis, outbreak prediction, and time series prediction of the percentage of respondents with positive COVID-19 tests and the percentage of respondents who show COVID-like illness. By clustering datasets across different demographics, we revealed micro and macro level insights into the relationship between symptoms and outbreaks of COVID-19. These insights might form the basis for future analysis of the epidemiology and manifestations of COVID-19 in different patient populations. Our correlation and prediction studies identified a small subset of features that can predict measures of COVID-19 prevalence to a high degree of accuracy. Using this, more efficient surveys can be designed to measure only the most relevant features to predict COVID-19 outbreaks. Shorter surveys will increase the likelihood of respondent participation and decrease the chances that respondents provide false (or incorrect) information. We believe that our analysis will be valuable in shaping health policy and in COVID-19 outbreak predictions for areas with low levels of testing by providing prediction models that rely on self-reported symptom data. As shown from our results, the predictions from our models could be reliably used by health officials and policymakers, in order to prioritize resources. Furthermore, having crowd-sourced information as the base helps scale this method at a much higher pace, if and when required in the future, e.g., due to the advent of a new virus or a strain.

In the future, we plan to use advanced deep learning models for predictions. Furthermore, given the promise shown by population level symptoms data, we find more relevant and timely problems that can be solved with individual data. Machine learning systems based on data from mobile/wearable devices can be built to understand users' vitals, sleep behavior, and so on. Having the data shared at an individual level can augment the participatory surveillance dataset and thereby the predictions made. This can be achieved without compromising the privacy of individuals. We also plan to compare the reliability of such survey methods with actual number of cases in the corresponding regions and its generalizability across populations.

Acknowledgement

We thank Seojin Jang, Chirag Samal, Nilay Shrivastava, Shrikant Kanaparti, Darshan Gandhi and Priyanshi Katiyar for their inputs in various stages of this study. We further thank Prof. Manuel Morales (University dé Montreal), Morteza Asgari and Hellen Vasques for helping in developing a dashboard to showcase the results. Lastly, we also thank Dr. Thomas C. Kingsley (Mayo Clinic) for his suggestions in the future works.

References

CDC. (2020). Data and statistics [Computer software manual]. (`https://www.cdc.gov/obesity/data/prevalence-maps.html`)

Chan, C. C., et al. (2020, Jun 02). Type i interferon sensing unlocks dormant adipocyte inflammatory potential. *Nature Communications*, *11*(1). Retrieved from `https://doi.org/10.1038/s41467-020-16571-4`

Cheung, Y.-W., & Lai, K. S. (1995). Lag order and critical values of the augmented dickey–fuller test. *Journal of Business & Economic Statistics*, *13*(3), 277–280.

Delphi group, C. M. U. (2020). *Delphi's covid-19 surveys.* Retrieved from `https://covidcast.cmu.edu/surveys.html`

Dong, E., Du, H., & Gardner, L. (2020). An interactive web-based dashboard to track covid-19 in real time. , *20*. doi: https://doi.org/https://doi.org/10.1016/S1473-3099(20)30120-1

Fan, J., et al. (2020). *Covid-19 world symptom survey data api.*

Friedman, J. H. (2001). Greedy function approximation: A gradient boostingmachine. *The Annals of Statistics*, *29*(5), 1189 – 1232. Retrieved from `https://doi.org/10.1214/aos/1013203451` doi: https://doi.org/10.1214/aos/1013203451

Galton, F. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, *15*, 246–263. Retrieved from `http://www.jstor.org/stable/2841583`

Garcia-Agundez, A., Ojo, O., Hernández-Roig, H. A., Baquero, C., Frey, D., Georgiou, C., ... others (2021). Estimating the covid-19 prevalence in

spain with indirect reporting via open surveys. *Frontiers in Public Health*, *9*.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with lstm. *1999 Ninth International Conference on Artificial Neural Networks ICANN 99.*.

Gostic, K., Gomez, A. C., Mummah, R. O., Kucharski, A. J., & Lloyd-Smith, J. O. (2020, February). Estimated effectiveness of symptom and risk screening to prevent the spread of covid-19. *eLife*, *9*. Retrieved from `https://europepmc.org/articles/PMC7060038` doi: https://doi.org/10.7554/elife.55570

Hale, T., Webster, S., Petherick, A., Phillips, T., & Kira, B. (2020). *Oxford covid-19 government response tracker blavatnik school of government.*

Holden, K. (1995). Vector auto regression modeling and forecasting. *Journal of Forecasting*, *14*(3), 159–166.

Liang, W., Liang, H., Ou, L., Chen, B., Chen, A., Li, C., ... for the China Medical Treatment Expert Group for COVID-19 (2020, 08). Development and Validation of a Clinical Risk Score to Predict the Occurrence of Critical Illness in Hospitalized Patients With COVID-19. *JAMA Internal Medicine*, *180*(8), 1081-1089. Retrieved from `https://doi.org/10.1001/jamainternmed.2020.2033` doi: https://doi.org/10.1001/jamainternmed.2020.2033

Menni, C., et al. (2020). Real-time tracking of self-reported symptoms to predict potential covid-19. *Nature medicine*, 1–4.

Menni, C., Valdes, A. M., Freidin, M. B., Sudre, C. H., Nguyen, L. H., Drew, D. A., ... Spector, T. D. (2020, Jul 01). Real-time tracking of self-reported symptoms to predict potential covid-19. *Nature Medicine*, *26*(7), 1037-1040. Retrieved from `https://doi.org/10.1038/s41591-020-0916-2` doi: https://doi.org/10.1038/s41591-020-0916-2

NIH. (2020). Adult body mass index (bmi) [Computer software manual]. (`https://www.nhlbi.nih.gov/health/educational/losewt/BMI/bmicalc.htm`)

Parma, V., et al. (2020). More than smell. covid-19 is associated with severe impairment of smell, taste, and chemesthesis. *medRxiv*. Retrieved from `https://www.medrxiv.org/content/early/2020/05/24/2020.05.04.20090902` doi: https://doi.org/10.1101/2020.05.04.20090902

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Quinlan, J. R. (1986, Mar 01). Induction of decision trees. *Machine Learning*, *1*(1), 81-106. Retrieved from `https://doi.org/10.1007/BF00116251` doi: https://doi.org/10.1007/BF00116251

Rodriguez, A., Muralidhar, N., Adhikari, B., Tabassum, A., Ramakrishnan, N., & Prakash, B. A. (2020). Steering a historical disease forecasting model under a pandemic: Case of flu and covid-19. *arXiv preprint arXiv:2009.11407*.

Rodriguez, A., Tabassum, A., Cui, J., Xie, J., Ho, J., Agarwal, P., ... Prakash, B. A. (2020). Deepcovid: An operational deep learning-driven frame-

work for explainable real-time covid-19 forecasting. *medRxiv*. Retrieved from https://www.medrxiv.org/content/early/2020/09/29/2020.09.28.20203109 doi: https://doi.org/10.1101/2020.09.28.20203109

Saad-Roy, C. M., et al. (2020). Immune life history, vaccination, and the dynamics of sars-cov-2 over the next 5 years. *Science*.

Shi, F., et al. (2020). Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. *IEEE Reviews in Biomedical Engineering*, 1–1. Retrieved from http://dx.doi.org/10.1109/RBME.2020.2987975 doi: https://doi.org/10.1109/rbme.2020.2987975

sklearn f regression [Computer software manual]. (2007-2020). (https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html)

Wilder, B., Mina, M. J., & Tambe, M. (2020). Tracking disease outbreaks from sparse data with bayesian inference. *arXiv preprint arXiv:2009.05863*.

Journal of Behavioral Data Science, 2021, *1* (1), 170–172.
DOI: https://doi.org/10.35566/jbds/v1n1/p7

Book Review: Mastering Software Development in R

Kévin Allan Sales Rodrigues[0000−0003−4925−5883]

University of São Paulo, São Paulo, Brazil
kevin@usp.br

Book review of **Mastering Software Development in R** by Roger D. Peng, Sean Kross and Brooke Anderson (2017). Victoria, British Columbia, Canada: Leanpub. 472 pages. Price $0.00 to $50.00 (e-book).
https://leanpub.com/msdr

The book **Mastering Software Development in R** is an excellent introduction to the use of R (R Core Team, 2020) software, and its focus is on teaching how to develop packages for R and how to create complex graphs with *ggplot2*. The package *ggplot2* is one of the most popular and downloaded R packages and was created in 2005 by Hadley Wickham as a data visualization package for the statistical programming language R based on the grammar of graphics–a general scheme for data visualization which breaks up graphs into semantic components such as scales and layers. In particular, *ggplot2* can serve as a replacement for the base graphics in R and contains a number of defaults for web and print display of data. The book covers a wide variety of other packages including: *choroplethr*, *choroplethrMaps*, *data.table*, *datasets*, *devtools*, *dlnm*, *dplyr*, *faraway*, *forcats*, *GGally*, *ggmap*, *ggthemes*, *ghit*, *GISTools*, *grid*, *gridExtra*, *httr*, *knitr*, *leaflet*, *lubridate*, *magrittr*, *methods*, *microbenchmark*, *package*, *pander*, *plotly*, *profvis*, *pryr*, *purrr*, *rappdirs*, *raster*, *RColorBrewer*, *readr*, *rmarkdown*, *scales*, *sp*, *stats*, *stringr*, *testthat*, *tidyr*, *tidyverse*, *tigris*, *titanic*, and *viridis*.

Developing R packages and making specialized statistical graphics are very relevant skills today because as new models and statistical methodologies emerge, there must be software available to apply the cutting edge theory to real problems. In addition, publishing packages on CRAN (comprehensive R archive network) is a way of scientific dissemination that can increase the impact of a scientific research because the packages allow for quick applications of the methodology developed in the research. The importance of data visualization is obvious and a well-thought-out graph can synthesize a lot of information (descriptive or inferential) clearly and intuitively.

The book has at least three main advantages: it is affordable (can even be acquired for free), it serves both as an introductory book for R and as a "bridge"

for more advanced books such as: Wickham (2015), Wickham (2016), Wickham (2019) and Xie, Allaire, and Grolemund (2018), and it goes straight to the point, allowing for faster and more fluid learning. It is an ideal book for anyone who wants to learn advanced R topics without investing a lot of time. I believe that this book is important for anyone who is starting to develop their own packages (including experienced researchers who are not familiar with programming or software development).

Now, I want to compare the book with four other books on R. Wickham (2015) is the reference book for anyone who wants to learn how to create their own R packages. It covers all the steps of creating a package, from organizing function codes to disseminating the package. Wickham (2016) is a book on *ggplot2* written by the main author of this package and consequently is a reference book on *ggplot2*. Wickham (2019) is a book that addresses more advanced R topics, such as metaprogramming and techniques to improve the performance of R codes. Xie et al. (2018) is the first official book authored by the core R Markdown developers that provides a comprehensive and accurate reference to the R Markdown ecosystem. The four books together cover the same content as the reviewed book and each was written by the package developers themselves or at least by people who have contributed a lot to the area corresponding to its content. The reviewed book is able to provide a broad overview of several important R topics but clearly does not offer the depth of a reference book. The reviewed book is great for beginning learners of the topics that are not covered in a first R course and the books mentioned here are useful if deeper understanding of any of the topics covered in the reviewed book is needed.

The book contains a brief introduction and 4 chapters with well-defined scopes. The introduction states the R packages that will be used in this book. Chapter 1 covers the introduction to R and how to clean and to tidy data. Chapter 2 covers introductory programming topics, such as if, else and object-oriented programming and other more advanced topics like profiling and benchmarking, robust error handling and debugging. Chapter 3 deals with building packages for R and covers R package development, writing good documentation and vignettes using *knitr* and *R Markdown*, writing tests for an R package using the *testthat* package, continuous integration[1] tools such as Travis and Appveyor, and distributing packages via CRAN and GitHub. Finally, Chapter 4 covers building graphics with the *ggplot2* package, creation of simple and dynamic maps, creation of new *ggplot2* theme by modifying an existing theme, creation of new geom function to implement a new feature or simplify a workflow, and other related topics.

Each of the book's 4 chapters begins with a description of what will be learned in a short paragraph and follows by a list of topics covered in the chapter. At the beginning of each section, there is also a list of topics covered, except for the sections of Chapter 4 and Sections 2.8, 3.1 and 3.10. These sections without a list

[1] The topic of continuous integration is a little known topic in the statistical community and has the role of ensuring that the package continues to function properly after successive updates.

of topics covered are either conceptual in nature or the title is self-explanatory. The list of topics covered helps navigate the book and learn exactly what you want without having to read the entire book. This makes the book to be a quick reference to find information in short time. Although the book does not include exercises, it constantly encourages the readers to experiment with variations of the codes presented and for that it is enough to copy, paste and edit the codes of the book itself in `R`. For readers who have already taken an introductory `R` course or acquired basic knowledge of `R` through practice, Chapters 3 and 4 will certainly be most interesting. Chapter 3 provides technical details for the development and publication of packages on CRAN and GitHub. Publishing packages on CRAN and GitHub are great ways to share R code with the scientific community, but it's also possible to simply share the package in a standardized way at one's company, university, or research institute. Chapter 4 teaches how to develop customized visualization tools through packages such as *ggplot2* and *ggmap*.

In general, the book achieved the proposed objectives of what? Some changes can be made to improve a reader's experience. For example, it can list the `R` packages required for each of the 4 chapters so the reader can prepare the computational environment in advance for a specific chapter, such as Chapter 3 or 4. It also benefits by adding an index to show the pages on which each package it is mentioned since the book has more than 400 pages. By doing so, the reader interested in using a specific package can be directed more quickly to the examples of the desired package.

Acknowledgements

The author gratefully acknowledge CNPq (Brazilian National Council for Scientific and Technological Development) for the financial support.

References

R Core Team. (2020). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `http://www.R-project.org`

Wickham, H. (2015). *R packages: Organize, test, document, and share your code.* O'Reilly Media, Inc.

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis.* Springer-Verlag.

Wickham, H. (2019). *Advanced r (2nd ed.).* CRC press.

Xie, Y., Allaire, J. J., & Grolemund, G. (2018). *R markdown: The definitive guide.* CRC Press.